UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/691,966	10/24/2003	Khader S. Abdel-Hafez	3359-Z	6973

7590 08/01/2007

Law Office of Jim Zegeer
Suite 108
801 North Pitt Street
Alexandria, VA 22314

EXAMINER	
TRIMMINGS, JOHN P	
ART UNIT	PAPER NUMBER
2117	

MAIL DATE	DELIVERY MODE
08/01/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/691,966
Filing Date: October 24, 2003
Appellant(s): ABDEL-HAFEZ ET AL.

Jim Zegeer
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 06/07/2007 appealing from the Office action mailed 02/16/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,166,604	AHANAN et al.	11-1992
2002/0120896	WANG et al.	8-2002

ABDEL-HAFEZ et al., "Applicant Admitted Prior Art", (herein AAPA), Figures 2A-E and Background of the Disclosure.

Lattice Semiconductor, "ORCA Series 4 FPGAs", (herein Lattice), Data Sheet, January, 2003.

(9) Grounds of Rejection

The following grounds of rejection are applicable to the appealed claims:

1. Domestic Priority Not Granted:

Appellant's claim for domestic priority under 35 U.S.C. 119(e) was acknowledged. However, the provisional application upon which priority was claimed failed to provide adequate support under 35 U.S.C. 112 for claims 83-106 of this application. The lack of support in the provisional application was in reference to global scan enable and global set/reset enable signals being used to control the set/reset controller.

Applicant's arguments filed 12/12/2005 in regard to the benefit of priority of the provisional application were fully considered but they were not persuasive. The later-

filed application must have been an application for a patent for an invention that was also disclosed in the prior application (the parent or original nonprovisional application or provisional application). The disclosure of the invention in the parent application and in the later-filed application must have been sufficient to comply with the requirements of the first paragraph of 35 U.S.C. 112. See *Transco Products, Inc. v. Performance Contracting, Inc.*, 38 F.3d 551, 32 USPQ2d 1077 (Fed. Cir. 1994). Neither requirements above were met.

The disclosure of the prior-filed provisional application, Application No. 60/422,117, failed to provide adequate support or enablement in the manner provided by the first paragraph of 35 U.S.C. 112 for one or more claims of this application. The appellant's argument in the amendment after-non-final stated that the prior-filed application "implied" global set/reset and scan-enable. The examiner maintained in the final rejection of 02/16/2006 that the Disclosure, failed to refer to any "global" signals and had not named any signals as being "global", and therefore failed to teach global set/reset and global scan-enable as was claimed in the two independent claims of the subject application. Since there was no teaching of the signals, the examiner maintained the finding in the previous office action dated 9/13/2005, that domestic priority based on the prior provisional application is not granted.

2. Rejection of claims 83-106:

Claims 83-106 were rejected under 35 U.S.C. 103(a) as being unpatentable over AHANIN et al., U.S. Patent No. 5166604 (herein Ahanin), in view of the appellant's

admitted prior art (herein AAPA), in view of WANG et al., U.S. Patent Application No. 2002/0120896 (herein Wang), and further in view of "ORCA® Series 4 FPGAs" by Lattice Semiconductor (herein Lattice).

As per independent claims 83 and 96:

Ahanin teaches a method for testing faults propagated to the data ports asynchronous set/reset ports of scan cells in an integrated circuit based on a set/reset controller (FIG.2 4) having a scan enable signal and a set/reset enable signal for testing faults propagated to the data ports and asynchronous set/reset ports of selected scan cells (see Background of Ahanin) in a scan-based integrated circuit (see FIG.2), the scan-based integrated circuit containing one or more set/reset circuitries (FIG.2 102, 112) and one or more scan chains (for example column 2 lines 22-52 and FIG.1), each scan chain comprising multiple scan cells coupled in series (see example of FIG.1), each scan cell having one or more clocks (FIG.2 12f, 12i); but Ahanin fails to further teach said set/reset controller comprising shift and capture controllers.

But in the AAPA, the features are disclosed as follows; a shift controller (AAPA FIG.2D 262), inserted between a selected set/reset circuitry (FIG.2D 203) and said asynchronous set/reset ports of all said selected scan cells (FIG.2D 205), for disabling said asynchronous set/reset ports of all said selected scan cells (FIG.2D 262), when said scan enable signal is enabled (in FIG.2D, the AND performs the function), during a shift-in or shift-out operation (SE used for shifting is obvious because it is well known in the art); and a capture controller (AAPA FIG.2E 281), inserted between said selected set/reset circuitry (FIG.2E 203) and said asynchronous set/reset ports of all said

selected scan cells (FIG.2E 205), for enabling or disabling propagation of said faults present in said selected set/reset circuitry to said asynchronous set/reset ports of all said selected scan cells, in response to said set/reset enable signal when said scan enable signal is disabled (in FIG.2E, the multiplexer 281 performs the function), during a capture operation (SE disablement is an obvious and well known occurrence during capture).

Motivation for the combination of the AAPA with Ahanin comes from the advantages stated in the applicant's Background statement on page 6, as advantageously detecting set/reset faults during scan testing and capture cycles. One with ordinary skill in the art at the time of the invention, motivated as suggested, would have found it obvious to include shift and capture circuits as described in the AAPA to the controls of Ahanin in order to improve set/reset detection capabilities in a scan circuit.

However, neither Ahanin nor the AAPA provide a disclosure wherein the set/reset enable and the scan enable signals are in fact global signals. But, a teaching that each signal is in fact disclosed as being global is incorporated into the analogous art references of Lattice (Global Set/Reset Enable) on pages 36 in Table 16, and page 61 column 1 last paragraph, and Wang (for example FIG.2 201, Global Scan Enable).

Motivation for Lattice boasts of new features and enhancements not available earlier (in the Introduction); and for Wang, in paragraph [0011], states the disadvantage of complex scan enable requirements supplanted by a simpler approach resulting in the advantageous use of a global scan enable (paragraph [0017]). One with ordinary skill in

the art at the time of the invention, motivated as suggested, would have found it obvious to incorporate the improvements of Lattice and Wang with Ahanin and the AAPA in order to enhance set/reset testing along with improved and simpler circuitry (a global signal) for scan enable.

NOTES:

The *appellant is silent* in this Appeal Brief in regard to a claim for a CIP based on application No. 10/067,372. Please note that in the final office action dated 02/16/2006, the appellant's claim for the benefit of a prior-filed application under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, or 365(c) *was not granted.* The reason for refusing priority was that the appellant did not comply with one or more conditions for receiving the benefit of an earlier filing date under 35 U.S.C. 119 or 120 as follows:

The application claimed the benefit of prior-filed nonprovisional application No. 10/067,372 (WANG) under 35 U.S.C. 120, 121, or 365(c). However, Rules, 37CFR 1.63e requires a newly executed oath ("A newly executed oath or declaration must be filed in any continuation-in-part application, which application may name all, more, or fewer than all of the inventors named in the prior application"). The appellant did not complied with said CFR 163e, therefore, the examiner did not enter the amendment to the Specification submitted by the appellant dated 12/12/2005, in which the applicant claimed that the instant application is a CIP of 10/067,372. Since *the appellant is silent* in regard to this action in re: WANG, the examiner maintains the non-entry of the claim for priority to 10/067,372 (WANG), and therefore WANG i*s kept as a reference for*

rejections in claims 83-106 (independent claims 83 and 96 in particular) in view of the publication date of 08/29/2002.

(10) Response to Argument

As to Ground No. 1:

Appellant's arguments filed in the brief in regard to the benefit of priority of the provisional application (pages 9 and 10 of the brief) have been fully considered but they are not persuasive. The later-filed application must be an application for a patent for an invention that is also disclosed in the prior application (the parent or original nonprovisional application or provisional application). The disclosure of the invention in the parent application and in the later-filed application must be sufficient to comply with the requirements of the first paragraph of 35 U.S.C. 112. See *Transco Products, Inc. v. Performance Contracting, Inc.*, 38 F.3d 551, 32 USPQ2d 1077 (Fed. Cir. 1994).

The disclosure of the prior-filed provisional application, Application No. 60/422,117, fails to provide adequate support or enablement in the manner required by the first paragraph of 35 U.S.C. 112 for one or more claims of this application. The appellant's argument that the prior-filed application "implies" global set/reset and global scan-enable without mentioning the word in any context of a global signal, or benefits of a global signal, is not sufficient to teach a global set/reset and a global scan-enable as is claimed in the two independent claims of the subject application. Since there is no teaching of the signals, the examiner maintains the finding in the previous office action

dated 9/13/2005, that domestic priority based on the prior provisional application is not granted.

As to Ground No. 2:

Appellant's arguments filed in the brief in regard to the rejections of claims 83-106 under 35 USC 103, in particular, the argument for the independent claims 83 and 96 (pages 10-13 of the brief), have been fully considered but they are not persuasive. The examiner's grounds for rejection, in section (9) above, are maintained.

The appellant, on pages 10-11, argues that the combination of Ahanin, the AAPA, Wang and Lattice is based on hindsight, but has failed to discuss any fault in the examiner's rationale to combine the references. Further, the appellant sought to note differences between the examiner's combination and the appellant's claims on page 11, but instead addressed a figure (FIG. 3) in Ahanin that *was not a subject* in the rejection (FIG. 2 was the circuit used in Ahanin), and so the argument is not germane to the subject. Additional arguments noting certain characteristics of the circuits on page 11 (yet not claimed) are not persuasive because the examiner has determined that the combination of the references herein performs the disclosed and claimed functions. An argument and recitation of the intended use of a claimed invention must result in a structural difference between the claimed invention and the combination of prior art in order to patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim.

In regard to the AAPA, on page 12 of the brief, the appellant argued that the circuit lacks features of the claims. Since the rejection is under 35 USC 103, it must be viewed as a combination of Ahanin, AAPA, Lattice and Wang, and so the *combination* of Ahanin and Wang and Lattice with the AAPA should *be considered* as they apply to each feature. The appellant's argument fails to consider the *combined circuit* and instead the argument selects portions of one reference that are indeed met by another reference. One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In regard to Lattice (page 12-13 of the brief), the appellant argued that the examiner "reached into Lattice" to reject the claims based on a "global" signal (set/reset). However, the appellant again has failed to note any deficient citation in the rationale on the part of the examiner in choosing Lattice, that may have denied the use of such global signals.

Finally, it is noted that the rejections of the independent claims (see section 9 above) are based further on global signals in Wang, which the appellant has not addressed. And so the rejections of the independent claims, and all dependent claims in view of Ahanin, AAPA, Lattice and Wang are maintained.

(11) Related Proceedings Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/John P Trimmings/
John P Trimmings, Examiner

Conferees:

Guy Lamarre GUY LAMARRE PRIMARY EXAMINER

/Lynne H Browne/
Lynne H Browne
Appeal Practice Specialist, TQAS
Technology Center 2100

US005166604A

United States Patent [19]

Ahanin et al.

[11] Patent Number: 5,166,604

[45] Date of Patent: Nov. 24, 1992

[54] METHODS AND APPARATUS FOR FACILITATING SCAN TESTING OF ASYNCHRONOUS LOGIC CIRCUITRY

[75] Inventors: Bahram Ahanin, Cupertino; Craig S. Lytle, Palo Alto; Ricky W. Ho, Sunnyvale, all of Calif.

[73] Assignee: Altera Corporation, San Jose, Calif.

[21] Appl. No.: 611,974

[22] Filed: Nov. 13, 1990

[51] Int. Cl.[5] G01R 31/28

[52] U.S. Cl. ..... 324/158 R; 371/22.3; 371/22.6

[58] Field of Search ..... 371/22.1, 22.2, 22.3, 371/22.6, 25.1; 324/158 R, 73.1

[56] References Cited

U.S. PATENT DOCUMENTS

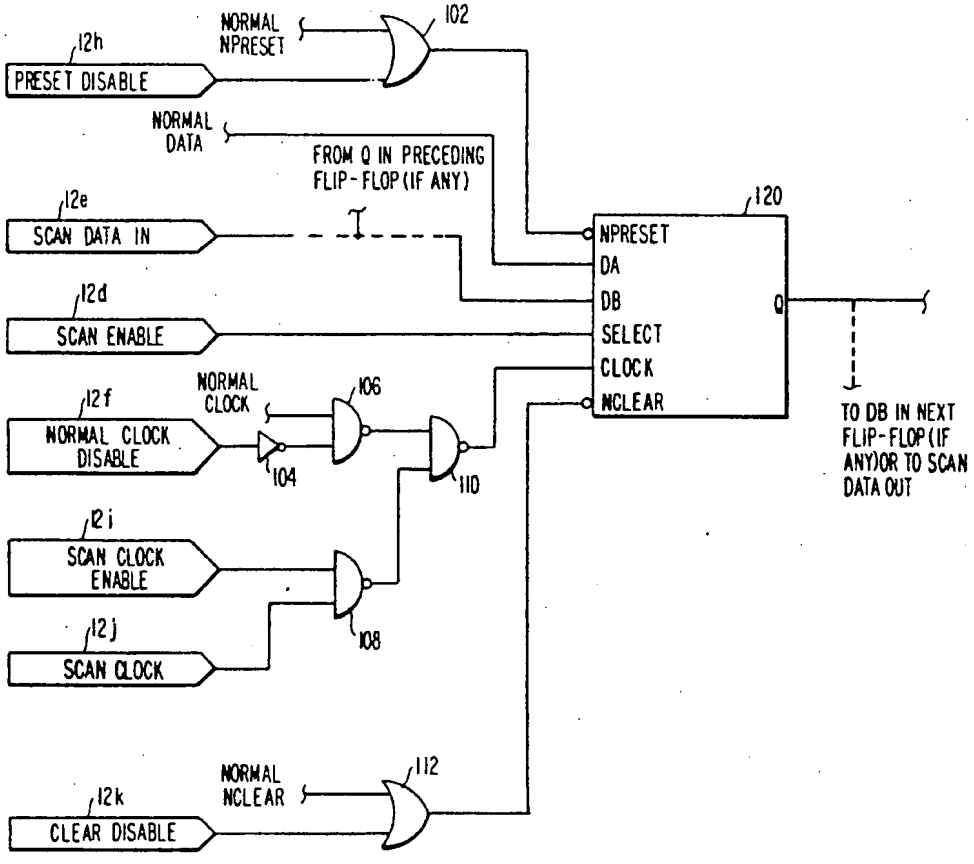
4,484,329	11/1984	Slamka et al.	371/25.1
4,556,840	12/1985	Russell	371/22.6
4,580,137	4/1986	Fiedler et al.	371/22.3
4,649,539	3/1987	Crain et al.	371/22.3
4,701,921	10/1987	Powell et al.	371/22.3
4,780,666	10/1988	Sakashita et al.	371/22.3
4,856,002	8/1989	Sakashita et al.	371/22.3
4,862,068	8/1989	Kawashima et al.	371/22.3
4,864,579	9/1989	Kishida et al.	371/22.3
5,008,618	4/1991	Van Der Star	371/22.3
5,047,710	9/1991	Mahoney	371/22.3

Primary Examiner—Ernest F. Karlsen
Assistant Examiner—William J. Burns
Attorney, Agent, or Firm—Robert R. Jackson; G. Victor Treyz

[57] ABSTRACT

Scan testing of asynchronous logic circuitry is facilitated by gating off the asynchronous inputs to flip-flops during scan testing. If desired, the asynchronous inputs which are gated off in this manner may themselves terminals or scan registers during testing. Alternatively, the asynchronous inputs which are gated could be tested by selectively enabling the signals at strategic points during scan testing. The number of input terminals required to control the test mode may be reduced by providing registers for storing test control signals applied to normal input terminals at the beginning of a test cycle. Once these test control signals are stored, the normal input terminals are free to return to their normal use.

10 Claims, 4 Drawing Sheets



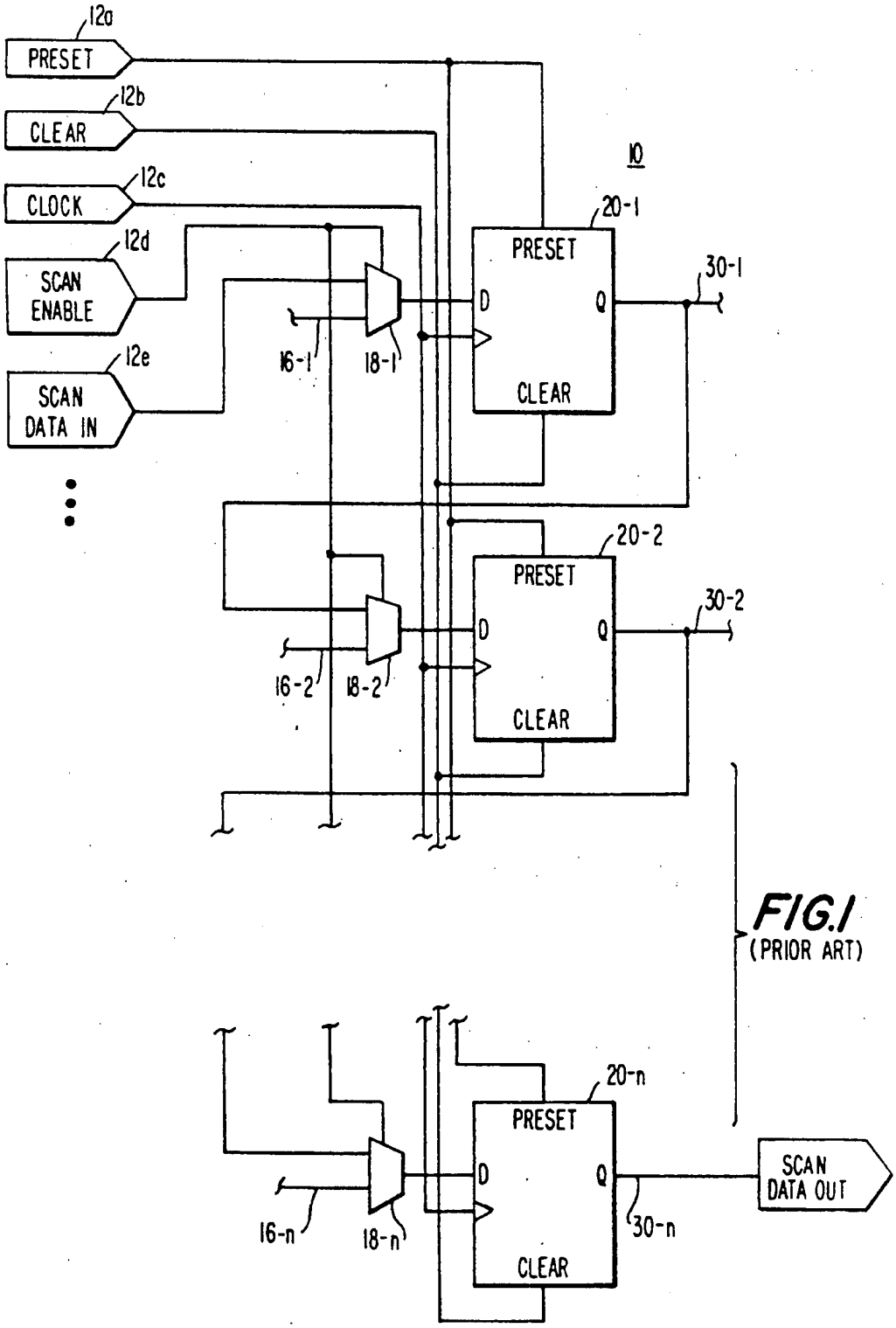


FIG.1
(PRIOR ART)

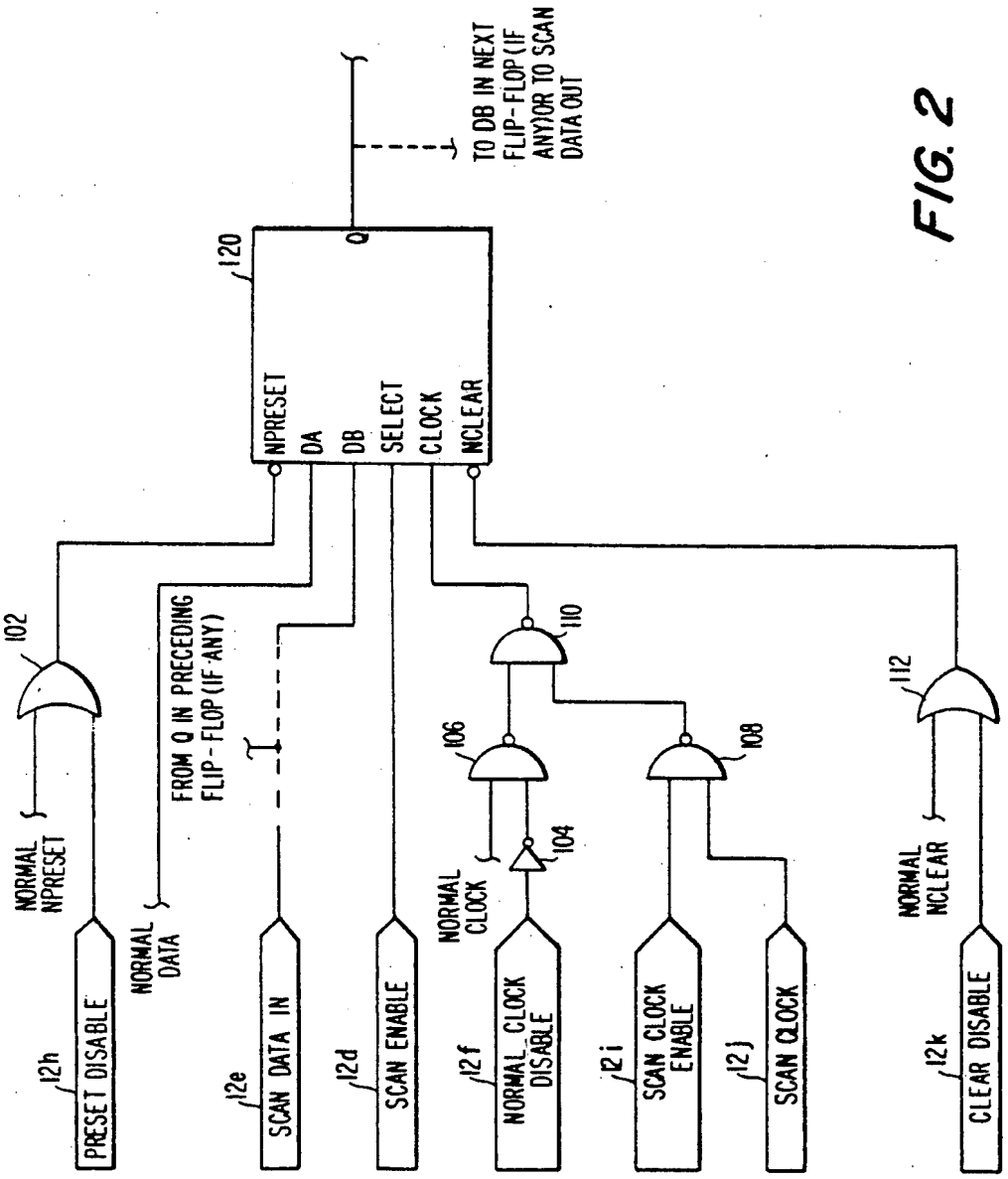

TO DB IN NEXT FLIP-FLOP (IF ANY) OR TO SCAN DATA OUT
Q
120
NPRESET
DA
DB
SELECT
CLOCK
NCLEAR
102
NORMAL NPRESET
PRESET DISABLE
12h
FROM Q IN PRECEDING FLIP-FLOP (IF ANY)
NORMAL DATA
SCAN DATA IN
12e
SCAN ENABLE
12d
106
110
NORMAL CLOCK
104
NORMAL_CLOCK DISABLE
12f
108
SCAN CLOCK ENABLE
12i
SCAN CLOCK
12j
112
NORMAL NCLEAR
CLEAR DISABLE
12k

FIG. 2

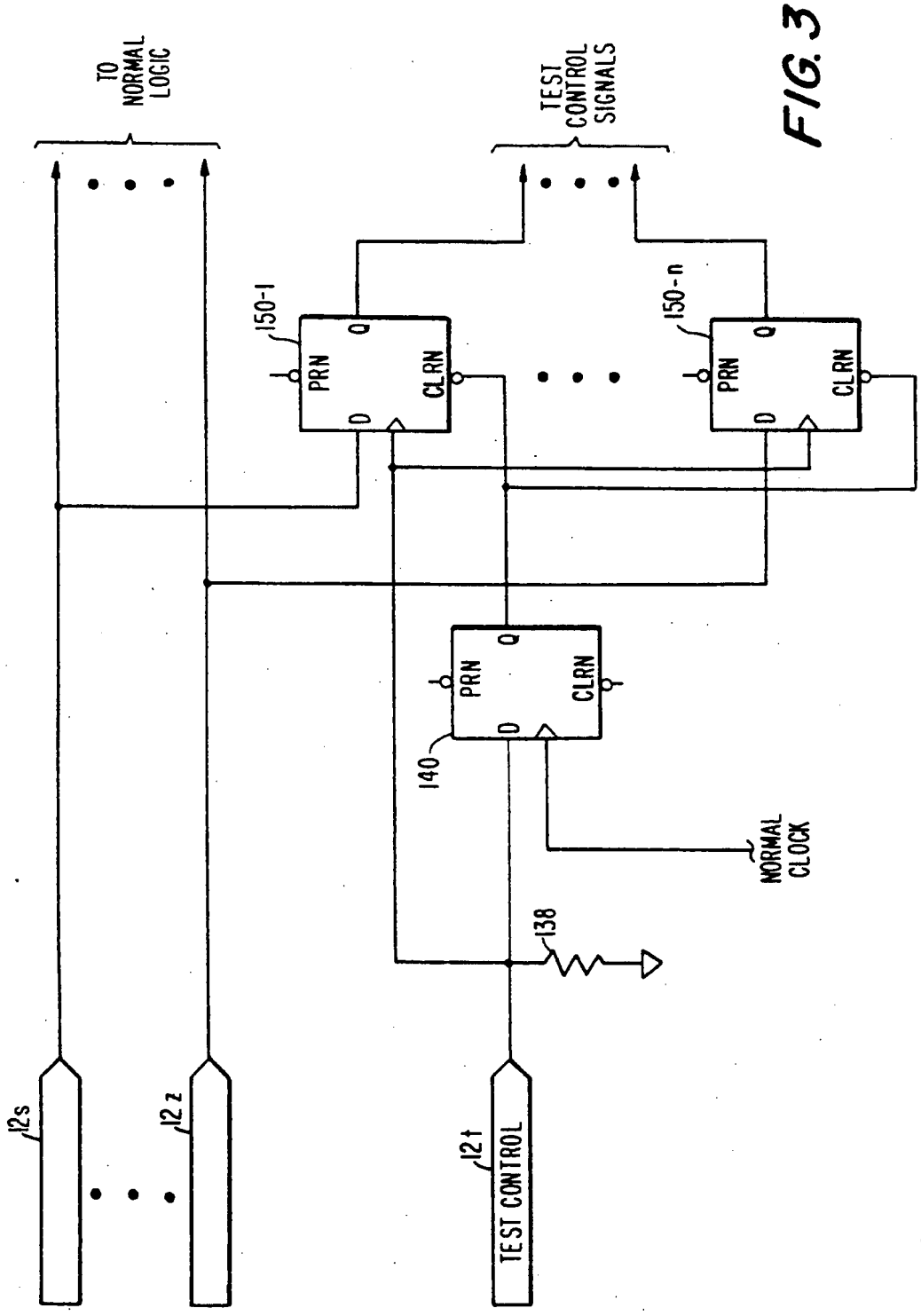

TO NORMAL LOGIC
TEST CONTROL SIGNALS
150-1
150-n
PRN
CLRN
Q
D
140
NORMAL CLOCK
138
12s
12z
12t
TEST CONTROL

FIG. 3

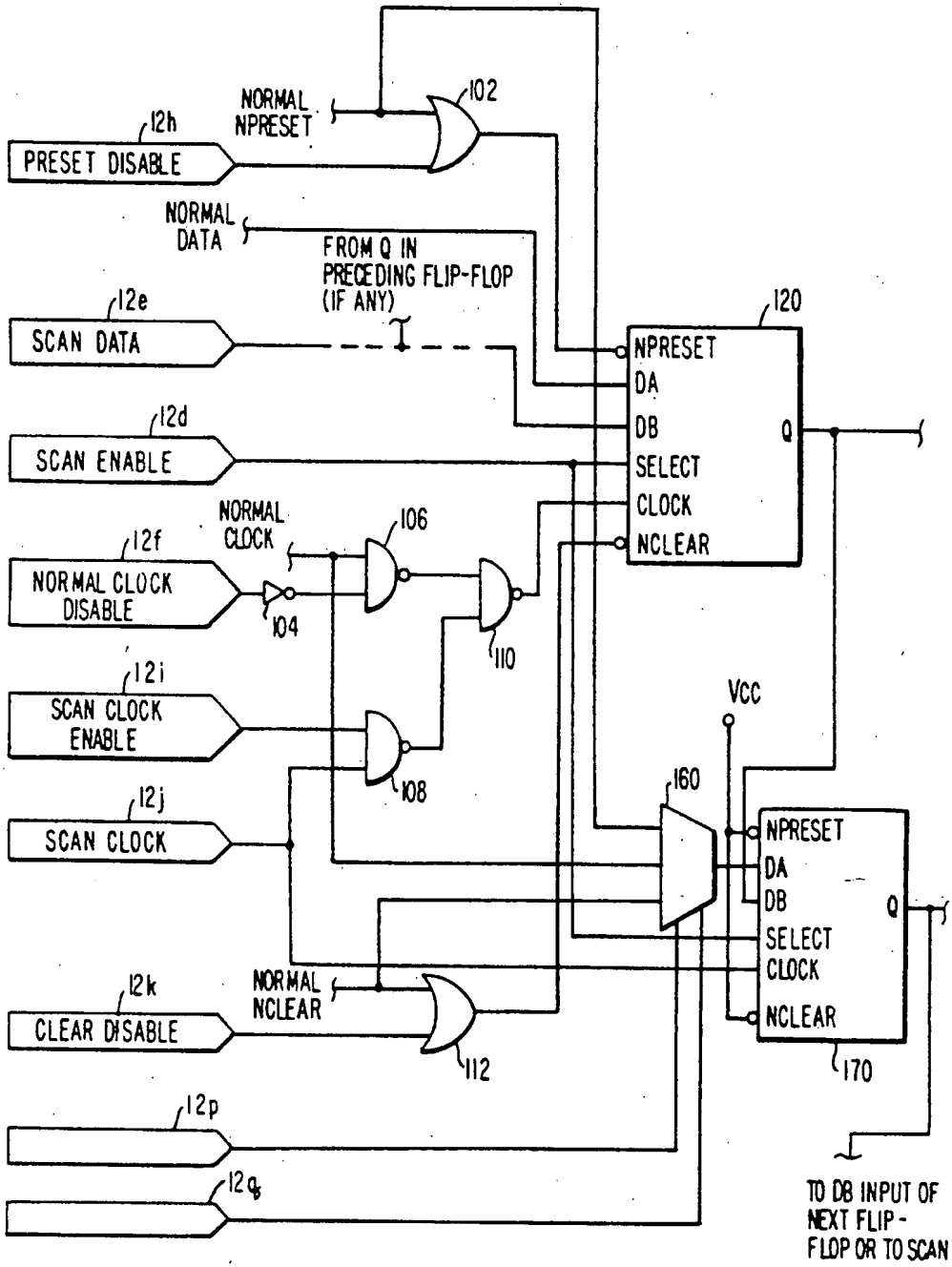

NORMAL NPRESET
102
12h
PRESET DISABLE
NORMAL DATA
FROM Q IN PRECEDING FLIP-FLOP (IF ANY)
12e
SCAN DATA
120
NPRESET
DA
DB
Q
SELECT
CLOCK
NCLEAR
12d
SCAN ENABLE
NORMAL CLOCK
106
12f
NORMAL CLOCK DISABLE
104
110
12i
SCAN CLOCK ENABLE
108
VCC
160
12j
SCAN CLOCK
NPRESET
DA
DB
Q
SELECT
CLOCK
NCLEAR
170
12k
CLEAR DISABLE
NORMAL NCLEAR
112
12p
12q
TO DB INPUT OF NEXT FLIP-FLOP OR TO SCAN DATA OUT TERMINAL

FIG. 4

METHODS AND APPARATUS FOR FACILITATING SCAN TESTING OF ASYNCHRONOUS LOGIC CIRCUITRY

BACKGROUND OF THE INVENTION

This invention relates to testing asynchronous logic circuitry, and more particularly to methods and apparatus for facilitating the testing of such circuitry.

Scan testing is a well-known technique for testing synchronous logic circuitry to determine whether or not the circuitry has the design required to enable it to function properly under all circumstances and/or whether the circuitry has been fabricated properly and without defects. However, scan testing assumes that signals such as the clock, preset, and clear of each storage register ("flip-flop") in the logic circuitry come directly from an input terminal (pin) of the circuitry. This is generally not true for asynchronous logic circuitry where one or more of such signals may be generated by logic included in the circuitry to be tested. This makes it impossible to test asynchronous logic either at all or as completely as would be desired using scan testing.

It is therefore an object of this invention to provide methods and apparatus for facilitating the scan testing of asynchronous logic.

It is another object of this invention to provide methods and apparatus for allowing more complete scan testing of asynchronous logic.

SUMMARY OF THE INVENTION

These and other objects of the invention are accomplished in accordance with the principles of the invention by modifying the design of asynchronous logic circuitry to be scan tested so that signals normally applied to such flip-flop inputs as the clock, preset, and clear inputs can be selectively gated off during scan testing of the logic circuitry. Additionally, if desired during testing, a scan clock signal can be applied to flip-flop clock inputs, and signals of predetermined logical value can be applied to flip-flop preset and clear inputs. Other circuit modifications may be made to further facilitate scan testing. For example, the circuitry may be modified so that few, one, or even no input pins must be dedicated to selecting and controlling the test mode. This can be done by providing one or more flip-flops for storing desired test mode control data applied to normal input pins when the circuitry is first put into test mode. Thereafter, this test mode control data can be read out of these flip-flops in order to free all or substantially all of the input pins for return to normal operation. To facilitate observation and testing of the signals which normally go to such destinations as flip-flop clock, preset, and clear inputs and which are gated off during scan testing, these signals may be applied to output terminals of the logic circuitry during testing, e.g., through one or more flip-flops added to the circuitry for this purpose. The number of such added flip-flops may be reduced by feeding signals to them through one or more multiplexers. An alternative technique for observing and testing the gated off signals is to selectively enable the signal and observe its effect on the scan logic.

Further features of the invention, its nature and various advantages will be more apparent from the accompanying drawings and the following detailed description of the preferred embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic block diagram of a representative portion of prior art synchronous logic circuitry which is arranged to enable scan testing of that logic circuitry.

FIG. 2 is a schematic block diagram of a representative portion of asynchronous logic circuitry including elements in accordance with this invention for facilitating scan testing of that logic circuitry.

FIG. 3 is a schematic block diagram of additional circuitry constructed in accordance with this invention which can optionally be used in conjunction with circuitry of the type shown in FIG. 2.

FIG. 4 is a schematic block diagram similar to FIG. 2 showing circuitry which can optionally be added to the FIG. 2 circuitry in accordance with this invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 shows synchronous logic circuitry 10 which has been conventionally adapted for scan testing. Each of flip-flops 20-1 through 20-n is part of the normal logic of circuitry 10 which is to be tested, generally along with other normal logic (not shown) which generates the data signals 16-1 through 16-n normally applied (via the associated one of multiplexers 18-1 through 18-n) to the data (D) input terminals of flip-flops 20, and/or which uses the registered (Q) output signals 30-1 through 30-n of those flip-flops. As is typical for synchronous logic, the preset, clear, and clock signals are respectively common for all of flip-flops 20 and respectively come directly from input terminals 12*a–c* of the circuitry. In addition, to enable scan testing of the circuitry, the Q output of each flip-flop 20 is connected to the other input terminal of the multiplexer 18 associated with the next flip-flop 20. Multiplexers 18 are controlled by a SCAN ENABLE input signal, and the second input terminal to the first multiplexer 18-1 is connected to a SCAN DATA IN terminal. This provides controllability of the circuit by allowing any desired scan data (entered via the SCAN DATA IN terminal when the SCAN ENABLE signal is high) to be shifted into flip-flops 20. After the desired scan data is shifted in, the normal logic data signals (16-1 through 16-n) can be tested by loading those signals into flip-flops 20 by bringing SCAN ENABLE low and applying the CLOCK signal. The registers 20 can then be observed by bringing SCAN ENABLE back high and shifting the registers out of the SCAN DATA OUT output pin.

The foregoing test method can generally not be used for asynchronous logic circuitry because in such logic circuitry some or all of the flip-flop clock, preset, and clear inputs may come from logic which is part of the circuitry, rather than simply from input pins 12 as in FIG. 1. FIG. 2 therefore shows how a typical module of the circuitry of FIG. 1 can be modified in accordance with the principles of this invention to facilitate scan testing exactly as discussed above, even though the module is part of asynchronous logic circuitry.

For simplicity in FIG. 2, a typical multiplexer 18 and the associated flip-flop 20 are combined in a single box 120 which will be referred to simply as flip-flop 120. Also for simplicity, the polarities of the preset and clear signals are reversed in FIG. 2, but those skilled in the art will understand that the polarities of all signals referred

to throughout this disclosure are arbitrary. In FIG. 2, the DA and DB signals correspond respectively to the two data signals applied to a typical multiplexer 18 in FIG. 1, and the SELECT signal in FIG. 2 corresponds to the control signal for that multiplexer in FIG. 1.

During normal operation of the apparatus shown in FIG. 2 the PRESET DISABLE, SCAN ENABLE, NORMAL CLOCK DISABLE, SCAN CLOCK ENABLE, and CLEAR DISABLE signals are all low. This allows OR gate 102 to apply the normal NPRESET signal (which may come from logic (not shown) which is part of the logic circuitry) to the NPRESET terminal of flip-flop 120. The normal data signal (also typically from logic (not shown) which is part of the logic circuitry) is applied to the DA terminal of flip-flop 120. The low SCAN ENABLE signal causes flip-flop 120 to select DA as its data input terminal. The low SCAN CLOCK ENABLE signal gates off the SCAN CLOCK signal, while the low NORMAL CLOCK DISABLE signal causes the normal clock signal (which again may come from logic (not shown) which is part of the logic circuitry) to pass through NAND gates 106 and 110 to the clock input terminal of flip-flop 120. Finally, the low CLEAR DISABLE signal allows OR gate 112 to apply the normal NCLEAR signal (which once again may come from logic (not shown) which is part of the logic circuitry) to the NCLEAR terminal of flip-flop 120.

To perform scan testing of the apparatus shown in FIG. 2, the PRESET DISABLE, SCAN ENABLE, NORMAL CLOCK DISABLE, SCAN CLOCK ENABLE, and CLEAR DISABLE signals may all be set high. This prevents the normal NPRESET and normal NCLEAR signals from affecting flip-flop 120. It also causes flip-flop 120 to select DB as its data input terminal. And it causes inverter 104 and NAND gate 106 to block the normal clock signal, while allowing NAND gates 108 and 110 to instead apply the SCAN CLOCK signal to the clock input terminal of flip-flop 120. SCAN DATA IN terminal 12*e* can now be used in conjunction with a SCAN CLOCK signal applied to input terminal 12*j* and the SCAN ENABLE signal applied to input terminal 12*d* to shift any desired test data into flip-flip 120 and the other similar flop-flops connected in a scan chain with it in the manner shown in FIG. 1. The FIG. 2 apparatus therefore allows scan testing of asynchronous logic in the same way that synchronous logic is scan tested in the prior art.

The embodiment shown in FIG. 2 requires a substantial number of input terminals (i.e., 12*d*, 12*e*, 12*f*, and 12*h–k*) to be dedicated to testing the logic circuitry. Input terminals are often a scarce resource in logic circuit design. One resolution in accordance with this invention is to tie some of the control input terminals together. This has the effect of reducing the number of input terminals required, while somewhat restricting the flexibility of the test logic. Those skilled in the art will appreciate that several combinations of connected inputs are possible and useful. This invention covers all of these combinations and is not limited to the notable cases of: (1) 12*h* and 12*k* connected; (2) 12*f* and 12*i* connected; or (3) 12*h*, 12*k*, 12*f*, and 12*i* connected.

FIG. 3 shows additional circuitry constructed in accordance with this invention which can also be used to dramatically reduce the required number of test input terminals. This is accomplished by allowing normal input terminals to receive test control signals which are then stored in test mode registers in the logic circuitry. These stored test control signals are then used to control the test, thereby allowing the input terminals which received them to return to normal use.

In the embodiment shown in FIG. 3 a single TEST CONTROL input terminal 12*t* is used to control whether or not the logic circuitry is in test mode. The other depicted input terminals 12*s–z* may be normal data input terminals of the logic circuitry. The data (D) input terminal of test control flip-flop 140 is normally held low by the signal applied to input terminal 12*t* and the connection of that terminal to ground through resistor 138. This keeps the Q output of flip-flop 140 low, which keeps test control signal storage registers 150-1 through 150-n in the reset state. Input terminals 12*s* through 12*z* can therefore be used as normal inputs to the normal logic of the device.

When it is desired to enter the test mode, the TEST CONTROL signal applied to input terminal 12*t* is raised to logical 1 and the desired test control signals (e.g., PRESET DISABLE, SCAN ENABLE, NORMAL CLOCK DISABLE, SCAN CLOCK ENABLE, and CLEAR DISABLE in the example shown in FIG. 2) are respectively applied to appropriate ones of input terminals 12*s* through 12*z*. The high TEST CONTROL signal causes the Q output of flip-flop 140 to rise after the next pulse on the normal clock. This allows the TEST CONTROL signal to clock flip-flops 150 so that each of these flip-flops stores a respective one of the test control signals applied to terminals 12*s* through 12*z*. Flip-flops 150 continue to store these signals as long as register 140 remains high. Moreover, the fact that the TEST CONTROL signal is constant prevents flip-flops 150 from receiving further data from input terminals 12*s* through 12*z*. Accordingly, after initially being used to supply the test control signals to flip-flops 150, terminals 12*s* through 12*z* can be returned to normal use during the balance of the test cycle.

When the test cycle is completed, the TEST CONTROL signal is returned to logical 0 which causes flip-flop 140 to reset all of flip-flops 150 at the next normal clock pulse.

Although the embodiment shown in FIG. 3 still requires one input terminal 12*t* dedicated to controlling the test mode, even this requirement can be eliminated by using a normally "illegal" combination of two or more normal input signals to indicate that the test mode is desired.

In FIG. 2 the asynchronous inputs to typical flip-flop 120 are gated off during scan testing. Among these gated off asynchronous signals are the normal NPRESET, CLOCK, and NCLEAR signals. With the use of traditional scan-logic circuitry and traditional scan-logic methodology, these gated off signals could not be tested because they do not feed a D-input to one of the scan flip-flops, and they must be constantly gated off during the scan testing. Nevertheless, for adequate testing of the logic circuitry it may be necessary or desirable to be able to observe one or more of these signals.

This invention provides two means of observing the faults at asynchronous inputs to the flip-flop. The first means is a direct result of the circuitry shown in FIG. 2. The second means involves some additional circuitry that will be discussed later.

FIG. 2 shows how each flip-flop's asynchronous inputs, NPRESET, CLOCK, and NCLEAR, are gated off before the flip-flop. It is an important aspect of this invention that the signals controlling the gating of these signals are discrete and can be enabled or disabled at

will. To observe the value of these asynchronous inputs all that need be done is to enable the controlling signals at a critical point in the scan testing. The enabled asynchronous input can immediately change the state of the flip-flop. When the flip-flop is later observed by shifting the scan flip-flops out via the SCAN DATA OUT pin, the result of the asynchronous input can be observed. Thus, by enabling various asynchronous inputs to the flip-flops one-by-one or many-at-once, the circuitry driving the asynchronous inputs to flip-flops can be observed.

Another method of observing the asynchronous inputs to the flip-flops in accordance with this invention is accomplished by applying each of these signals to an input terminal of one or more extra flip-flops added to the existing scan chain. In addition, to reduce the number of such extra flip-flops required to observe several signals, some or all of those signals may be applied to a multiplexer so that the signals can be applied to a single flip-flop one after another.

This aspect of the invention is illustrated by FIG. 4. All the elements of FIG. 2 are repeated in FIG. 4 with the addition of illustrative elements for allowing the normal NPRESET, CLOCK, and NCLEAR signals to be read out via multiplexer **160** and flip-flop **170** as part of the existing scan chain. Multiplexer **160** is controlled by signals from input terminals **12***p* and **12***q* to apply a selected one of the signals applied to the multiplexer to one of the data input terminals (DA) of flip-flop **170**. (Flip-flop **170** may be a device similar to device **120**.) The DA input of flip-flop **170** is selected when the SCAN ENABLE signal (applied to the SELECT input terminal of flip-flop **170**) is logical 0. Flip-flop **170** is clocked by the SCAN CLOCK signal applied to its clock input terminal. The NPRESET and NCLEAR input terminals of flip-flop **170** are tied to VCC (logical 1) to prevent presetting or clearing of the device. Accordingly, during scan testing, flip-flop **170** stores the signal applied to its DA input terminal in response to a SCAN CLOCK pulse and applies this stored signal to the next flip-flop in the scan chain, if one exists, or otherwise to the SCAN DATA OUT output terminal. In this way, each of the signals applied to multiplexer **160** can be observed one after another in successive SCAN CLOCK periods by changing the multiplexer control signals applied to input terminals **12***p* and **12***q*.

Those skilled in the art will appreciate that the use of multiplexer **160** is optional, and that separate flip-flops **170** could alternatively be provided for each of the signals applied to multiplexer **160**. Similarly, if a multiplexer **160** is provided, it can have any size (e.g., 2 to 1, 4 to 1, 8 to 1, 16 to 1, etc.)

It will be understood that the foregoing is merely illustrative of the principles of this invention, and that various modifications can be made by those skilled in the art without departing from the scope and spirit of the invention. For example, the number of gated-off signals applied to multiplexer **160** in FIG. 4 can be varied depending on which of those signals it is desired to be able to observe during testing.

What is claimed is:

1. Apparatus for facilitating testing of logic circuitry which includes storage register means having at least one asynchronous input selected from the group consisting of a preset input, a clock input, and a clear input, said asynchronous input being normally supplied with a first input signal produced by first logic circuit means, said apparatus comprising:

means for selectively inhibiting application of said first input signal to said asynchronous input when said logic circuitry is to be tested, wherein said means for selectively inhibiting is controlled by a control signal;

means for selectively storing an applied test control input signal and for producing an output signal indicative of the stored signal, said output signal being said control signal; and

means for enabling said means for selectively storing in order to store said test control input signal when said logic circuitry is to be tested, wherein said logic circuitry includes at least one normal data signal input terminal, wherein the signal applied to said input terminal may be stored in said storage register means during normal operation of said logic circuitry, and wherein the signal applied to said input terminal is also used as said test control input signal.

2. The apparatus defined in claim **1** wherein said means for selectively inhibiting comprises:

means for selectively applying a signal having a predetermined logical value to said asynchronous input in lieu of said first input signal.

3. The apparatus defined in claim **1** wherein said means for selectively inhibiting comprises:

controllable gate means interposed in circuit relationship between said first logic circuit means and said asynchronous input.

4. The apparatus defined in claim **3** wherein said controllable gate means logically combines said first input signal with a control signal so that said control signal can selectively override said first input signal.

5. The apparatus defined in claim **1** wherein said asynchronous input must be switchable both during normal operation of said logic circuitry and during testing of said logic circuitry, and wherein said means for selectively inhibiting comprises:

a source of a test switching signal; and

means for selectively applying said test switching signal to said asynchronous input in lieu of said first input signal.

6. The apparatus defined in claim **5** wherein said asynchronous input is said clock input.

7. The apparatus defined in claim **1** wherein said apparatus further comprises:

means for clearing the stored signal when said logic circuitry is no longer being tested.

8. Apparatus for facilitating testing of logic circuitry which includes storage register means having at least one asynchronous input selected from the group consisting of a present input, a clock input, and a clear input, said asynchronous input being normally supplied with a first input signal produced by first logic circuit means, said apparatus comprising:

means for selectively inhibiting application of said first input signal to said asynchronous input when said logic circuitry is to be tested;

an output terminal; and

means for applying said first input signal to said output terminal when said logic circuitry is to be tested, wherein said means for applying comprises:

data storage means for storing data applied to a data input terminal and applying the stored data to said output terminal; and

means for connecting said first input signal to said data input terminal when said logic circuitry is to be tested, such that said first input signal may be

observed when said logic circuitry is to be tested by connecting said output terminal to an existing scan chain.

9. The apparatus defined in claim 8 wherein a scan clock signal is employed during testing of said logic circuitry, wherein said data storage means has a clock input terminal and stores data applied to said data input terminal in response to the application of a clock signal to said clock input terminal, and wherein said scan clock signal is applied to said clock input terminal.

10. The apparatus defined in claim 8 wherein said first input signal is one of a plurality of signals which are inhibited from application to said storage register means during testing of said logic circuitry, and wherein said means for connecting comprises:

means for selectively applying any one of said plurality of signals to said data input terminal.

* * * * *

US 20040153926A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2004/0153926 A1**

Abdel-Hafez et al. (43) **Pub. Date: Aug. 5, 2004**

(54) **METHOD AND APPARATUS FOR TESTING ASYNCHRONOUS SET/RESET FAULTS IN A SCAN-BASED INTEGRATED CIRCUIT**

(76) Inventors: **Khader S. Abdel-Hafez**, San Francisco, CA (US); **Laung-Terng Wang**, Sunnyvale, CA (US); **Augusli Kifli**, Hsinchu (TW); **Fei-Sheng Hsu**, Hsinchu (TW); **Xiaoqing Wen**, Sunnyvale, CA (US); **Meng-Chyi Lin**, Taoyuan (TW); **Hsin-Po Wang**, Hsinchu (TW)

Correspondence Address:
Law Office of Jim Zegeer
Suite 108
801 North Pitt Street
Alexandria, VA 22314 (US)

(21) Appl. No.: **10/691,966** (Application) AAPA

(22) Filed: **Oct. 24, 2003**

Related U.S. Application Data

(60) Provisional application No. 60/422,117, filed on Oct. 30, 2002.

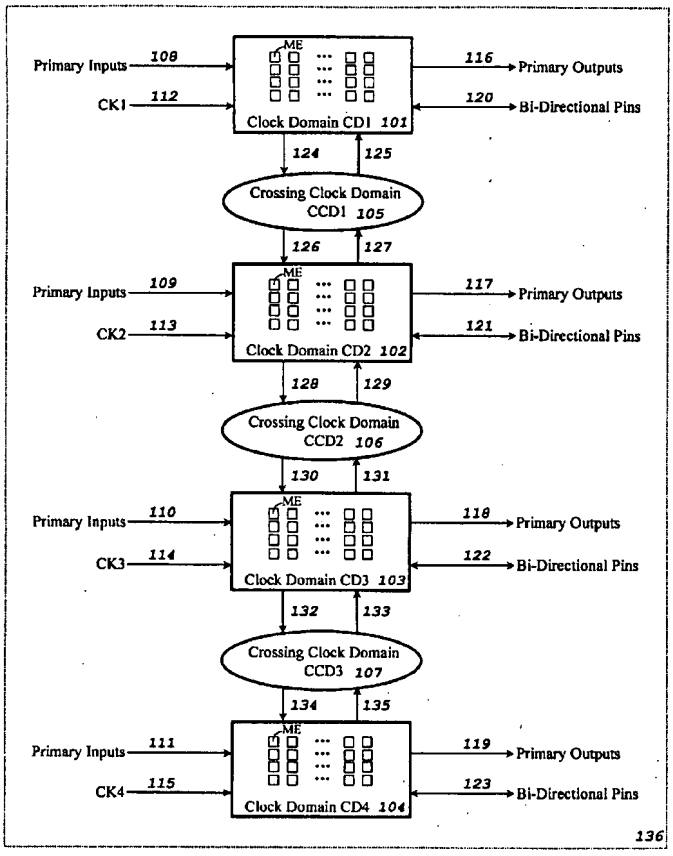
Publication Classification

(51) **Int. Cl.**[7] **G01R 31/28**

(52) **U.S. Cl.** **714/726**

(57) **ABSTRACT**

A method and apparatus to test data and set/reset faults in a scan-based integrated circuit in a selected scan-test mode or self-test mode. The scan-based integrated circuit contains multiple scan chains, each scan chain comprising multiple scan cells coupled in series. The method comprises shifting in a plurality of predetermined stimuli during scan-test or pseudo-random stimuli during self-test to the scan-based integrated circuit, using a set/reset enable (SR_EN) signal **383** and a scan enable (SE) signal **382** to capture faults to each scan cell, and shifting out the test responses for comparison or compaction. The apparatus or set/reset controller **375** further comprises using the set/reset enable (SR_EN) signal **383** and scan enable (SE) signal **382** to selectively propagate data faults or set/reset faults to the scan cells in the integrated circuit. Computer-aided design (CAD) methods are then proposed to automatically repair all asynchronous set/reset signals in the scan-based integrated circuit and generate test patterns comprising stimuli and test responses for verifying the correctness of the repaired scan-based integrated circuit during scan-test or self-test.



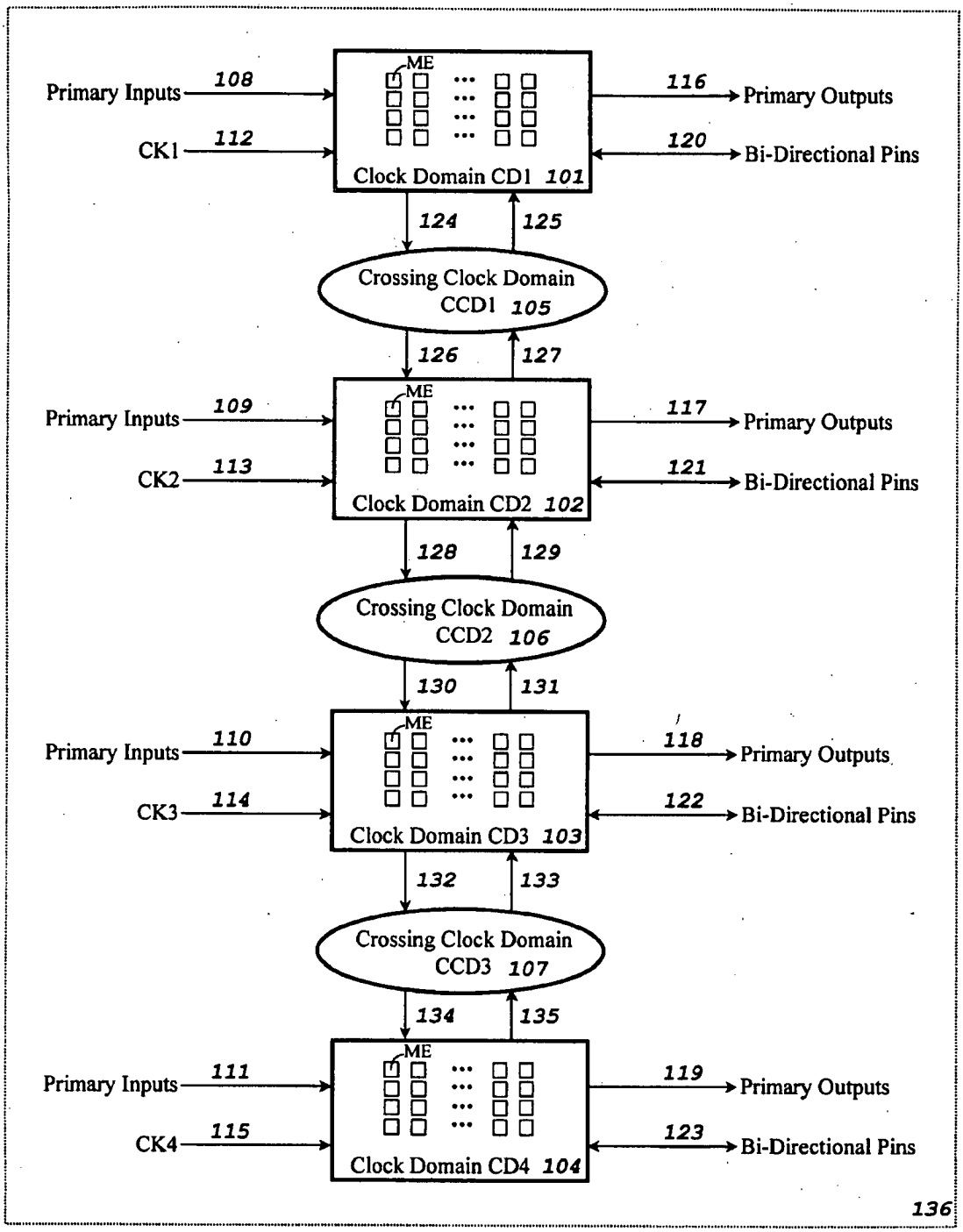

Primary Inputs
108
CK1
112
ME
Clock Domain CD1 101
116
Primary Outputs
120
Bi-Directional Pins
124
125
Crossing Clock Domain CCD1 105
126
127
Primary Inputs
109
CK2
113
ME
Clock Domain CD2 102
117
Primary Outputs
121
Bi-Directional Pins
128
129
Crossing Clock Domain CCD2 106
130
131
Primary Inputs
110
CK3
114
ME
Clock Domain CD3 103
118
Primary Outputs
122
Bi-Directional Pins
132
133
Crossing Clock Domain CCD3 107
134
135
Primary Inputs
111
CK4
115
ME
Clock Domain CD4 104
119
Primary Outputs
123
Bi-Directional Pins
136

FIG. 1A

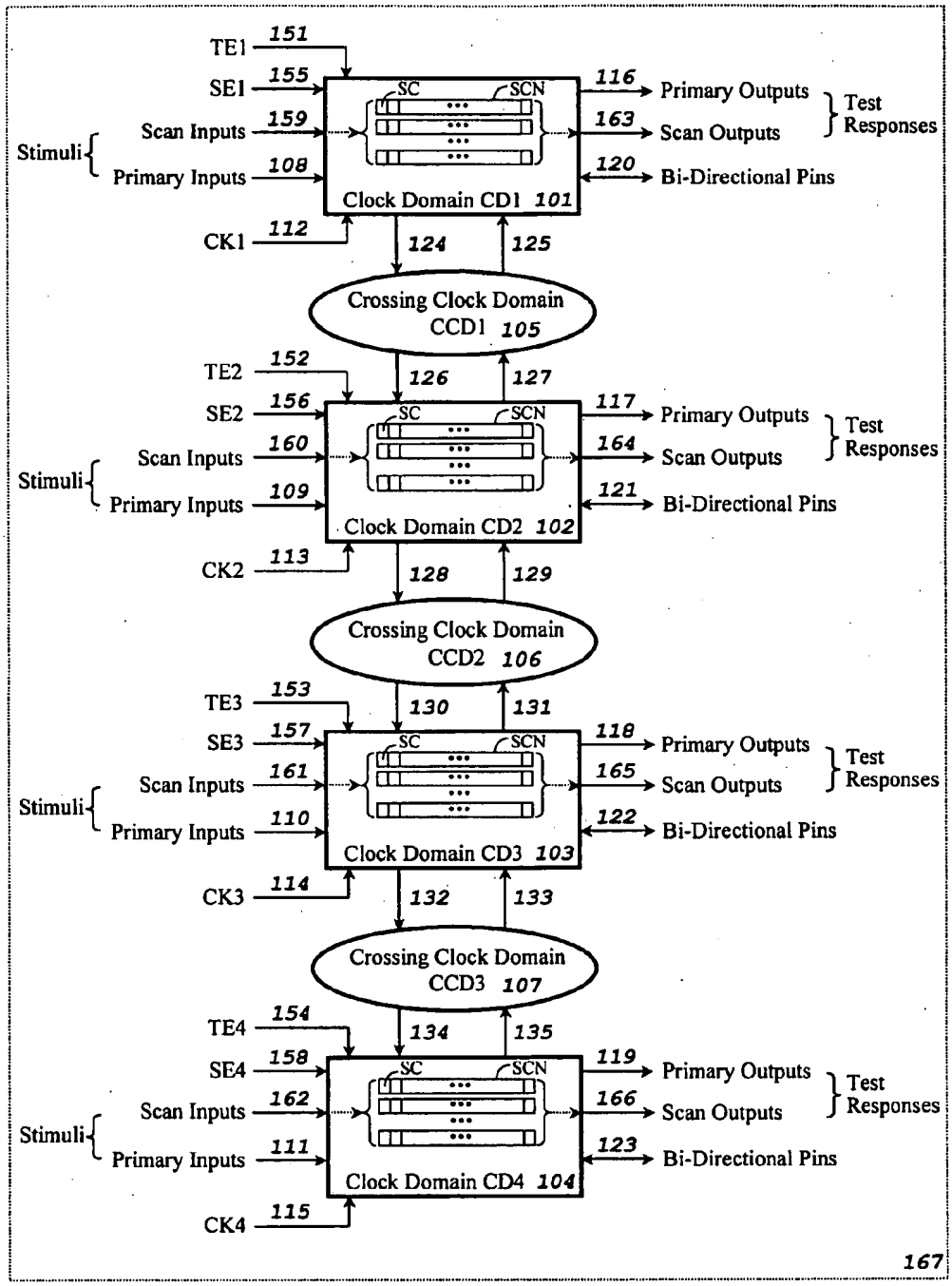

TE1
151
SE1
155
Stimuli
Scan Inputs
159
Primary Inputs
108
SC
SCN
116
Primary Outputs
163
Scan Outputs
120
Bi-Directional Pins
Test Responses
Clock Domain CD1 101
CK1
112
124
125
Crossing Clock Domain CCD1 105
TE2
152
SE2
156
Stimuli
Scan Inputs
160
Primary Inputs
109
117
Primary Outputs
164
Scan Outputs
121
Bi-Directional Pins
Test Responses
Clock Domain CD2 102
CK2
113
126
127
128
129
Crossing Clock Domain CCD2 106
TE3
153
SE3
157
Stimuli
Scan Inputs
161
Primary Inputs
110
118
Primary Outputs
165
Scan Outputs
122
Bi-Directional Pins
Test Responses
Clock Domain CD3 103
CK3
114
130
131
132
133
Crossing Clock Domain CCD3 107
TE4
154
SE4
158
Stimuli
Scan Inputs
162
Primary Inputs
111
119
Primary Outputs
166
Scan Outputs
123
Bi-Directional Pins
Test Responses
Clock Domain CD4 104
CK4
115
134
135
167

FIG. 1B

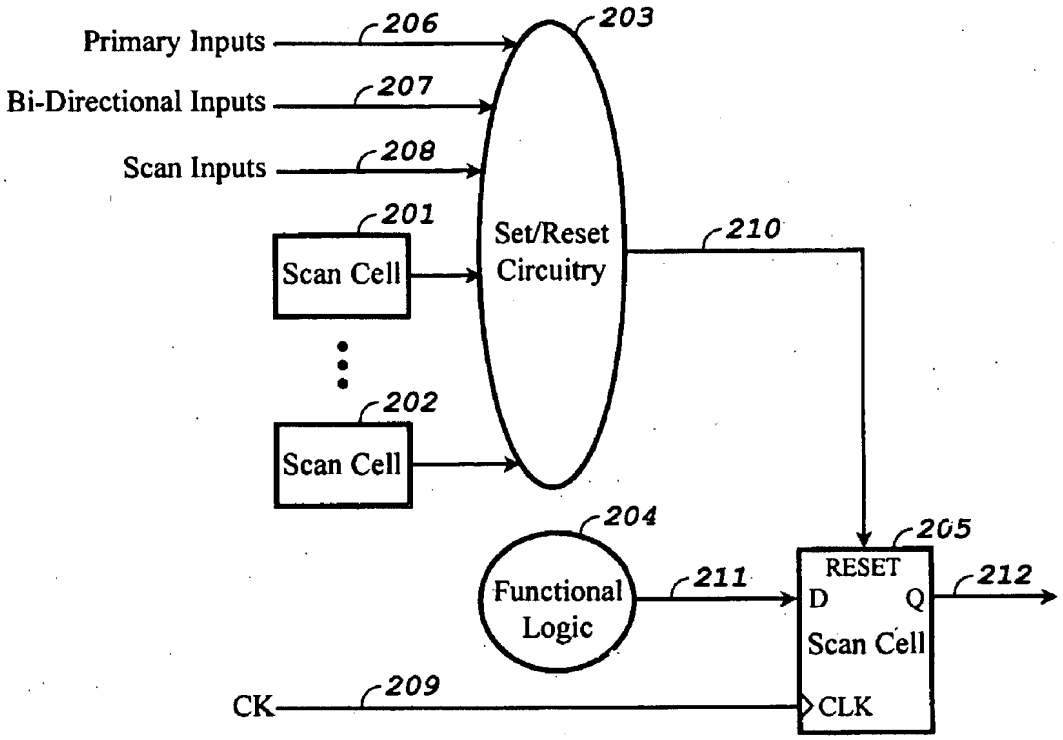
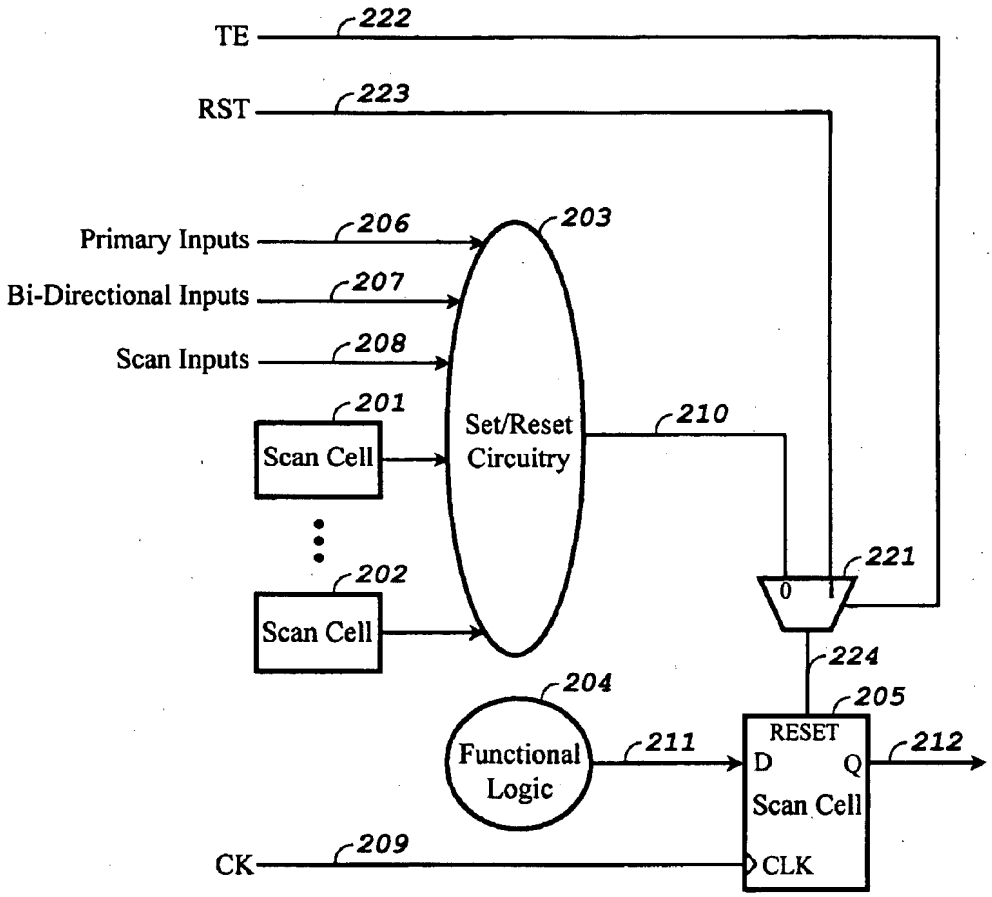
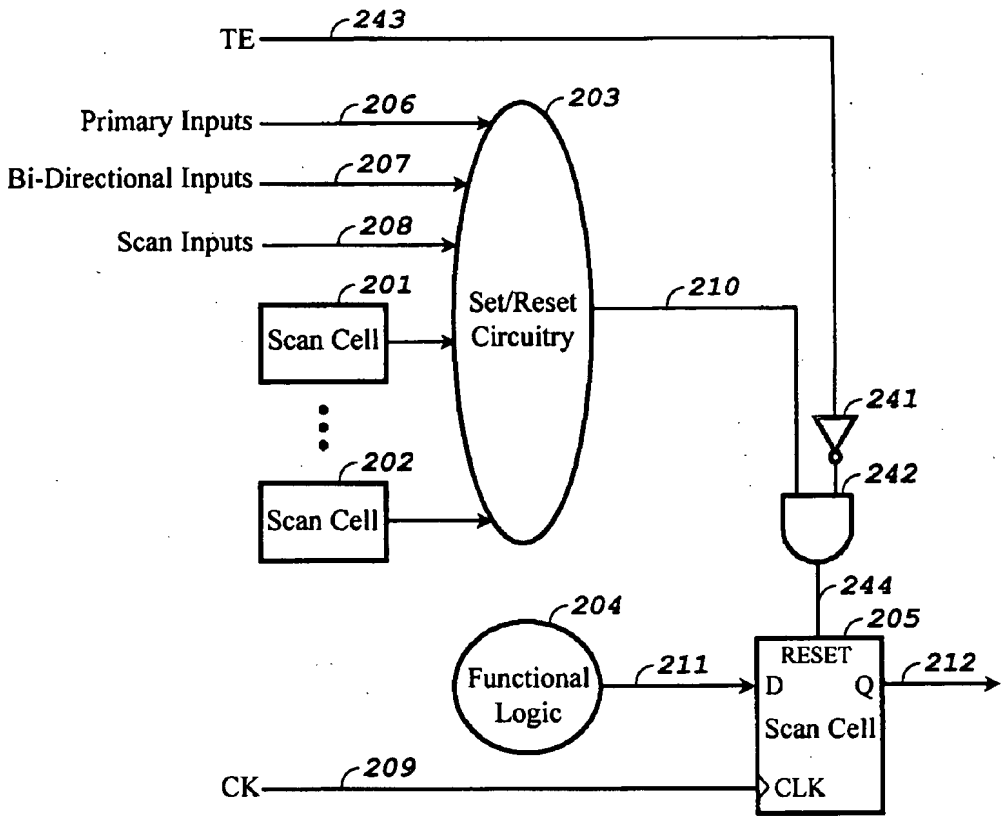

Primary Inputs
206
203
Bi-Directional Inputs
207
Scan Inputs
208
201
Scan Cell
Set/Reset
Circuitry
210
202
Scan Cell
204
205
RESET
Functional
Logic
211
D
Q
212
Scan Cell
CK
209
CLK

FIG. 2A



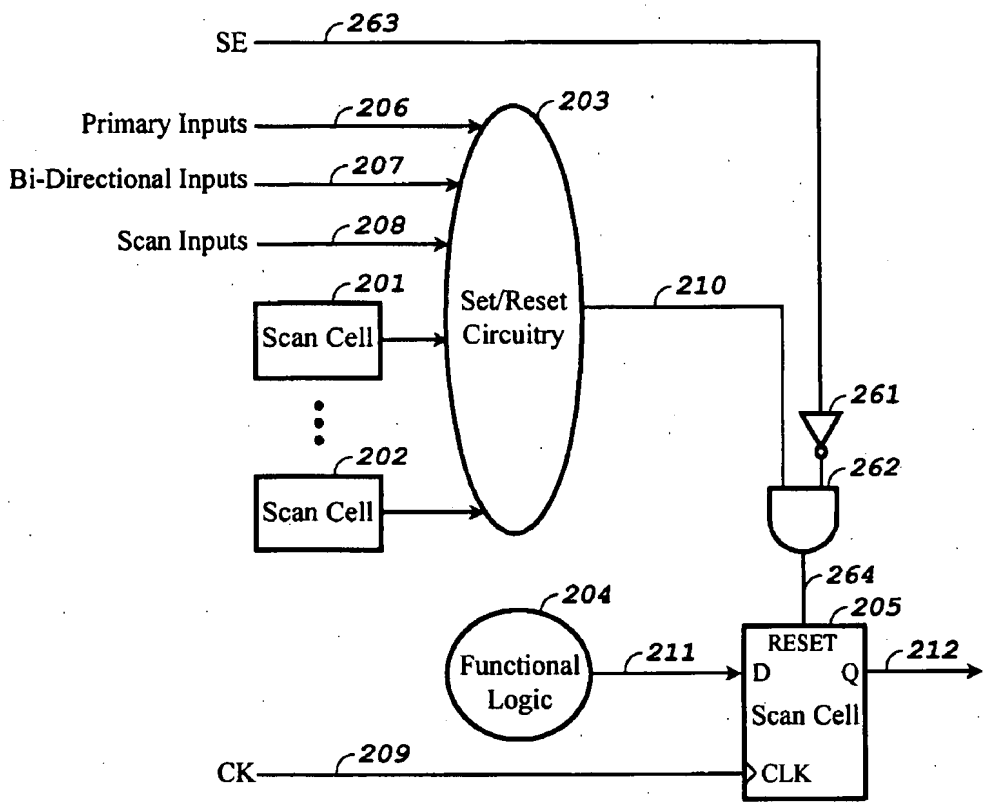
PRIOR-ART #1

FIG. 2B



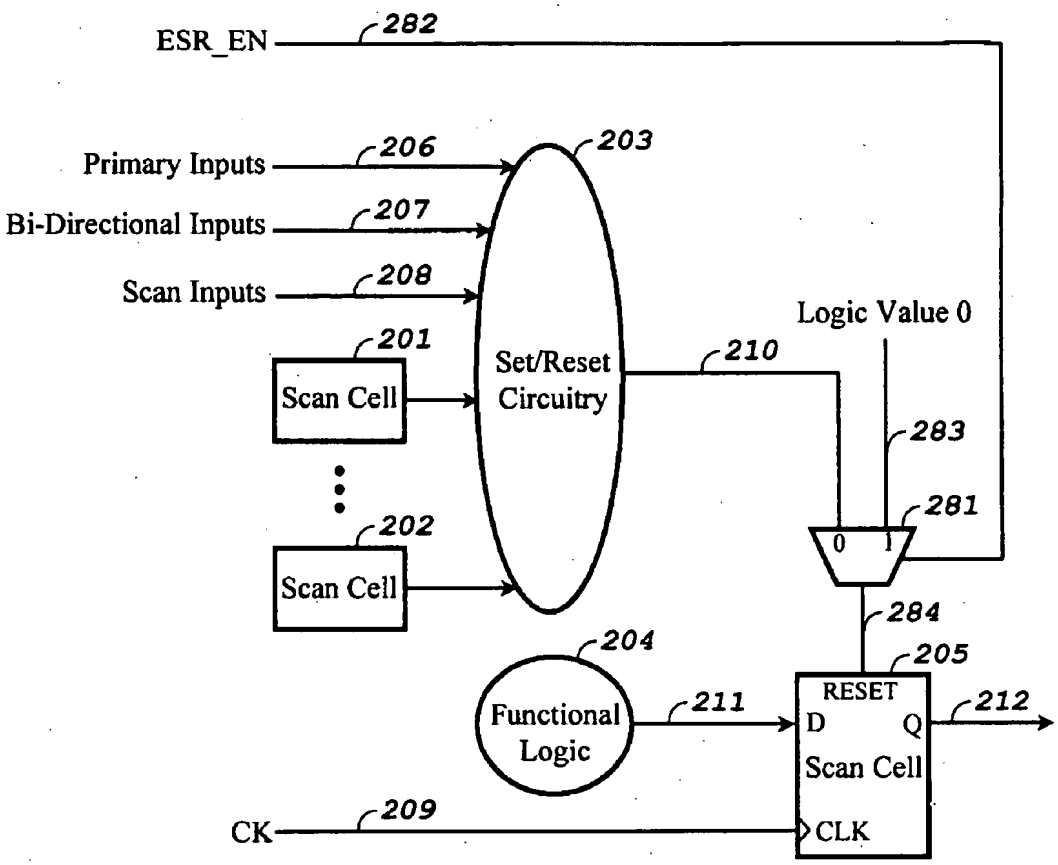
PRIOR-ART #2

FIG. 2C



PRIOR-ART #3

FIG. 2D



PRIOR-ART #4

FIG. 2E

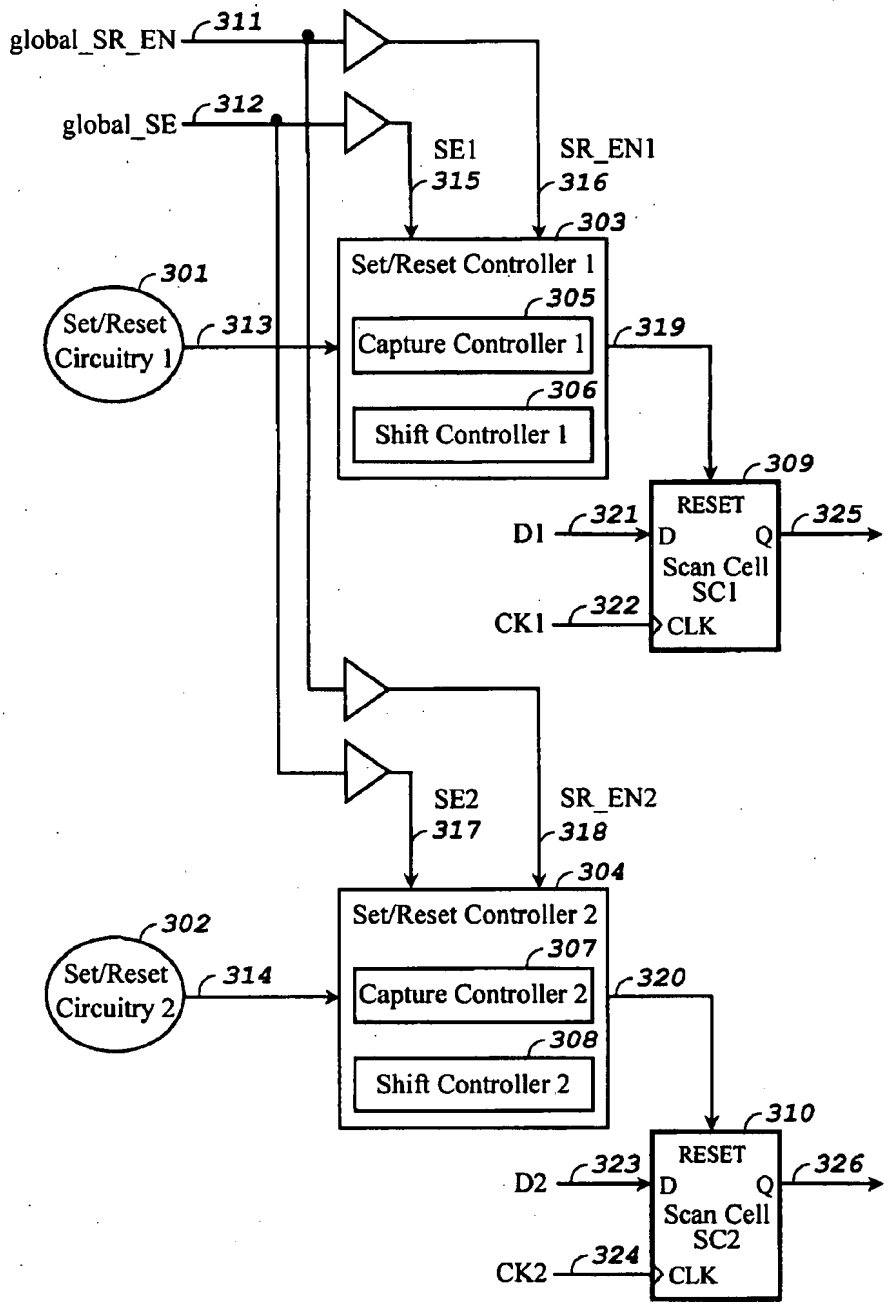

global_SR_EN
311
global_SE
312
SE1
315
SR_EN1
316
303
Set/Reset Controller 1
305
Capture Controller 1
306
Shift Controller 1
301
Set/Reset Circuitry 1
313
319
309
RESET
D
Q
Scan Cell SC1
CLK
D1
321
325
CK1
322
SE2
317
SR_EN2
318
304
Set/Reset Controller 2
307
Capture Controller 2
308
Shift Controller 2
302
Set/Reset Circuitry 2
314
320
310
RESET
D
Q
Scan Cell SC2
CLK
D2
323
326
CK2
324

FIG. 3A

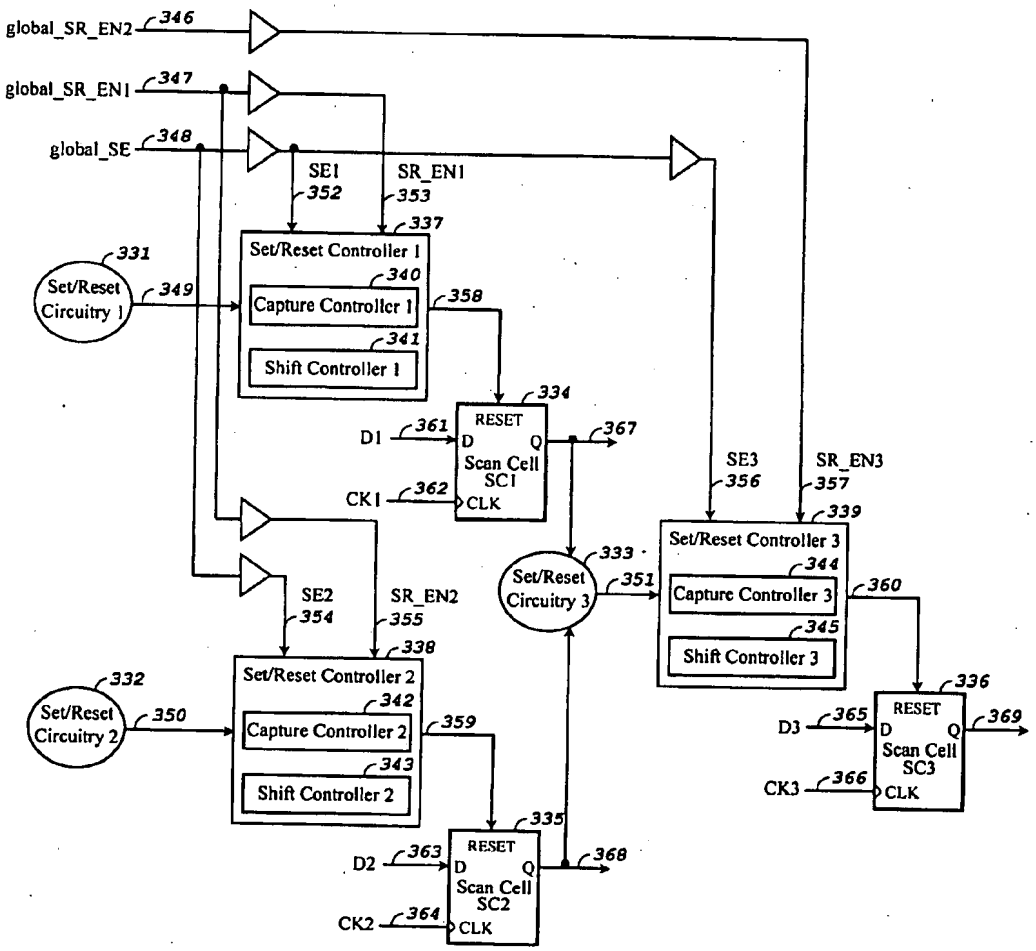

global_SR_EN2
346
global_SR_EN1
347
global_SE
348
SE1
352
SR_EN1
353
337
Set/Reset Controller 1
340
Capture Controller 1
341
Shift Controller 1
331
Set/Reset Circuitry 1
349
358
334
RESET
D
Q
Scan Cell SC1
CLK
D1
361
367
CK1
362
SE3
356
SR_EN3
357
339
Set/Reset Controller 3
344
Capture Controller 3
345
Shift Controller 3
333
Set/Reset Circuitry 3
351
360
SE2
354
SR_EN2
355
338
Set/Reset Controller 2
342
Capture Controller 2
343
Shift Controller 2
332
Set/Reset Circuitry 2
350
359
336
RESET
D
Q
Scan Cell SC3
CLK
D3
365
369
CK3
366
335
RESET
D
Q
Scan Cell SC2
CLK
D2
363
368
CK2
364

FIG. 3B

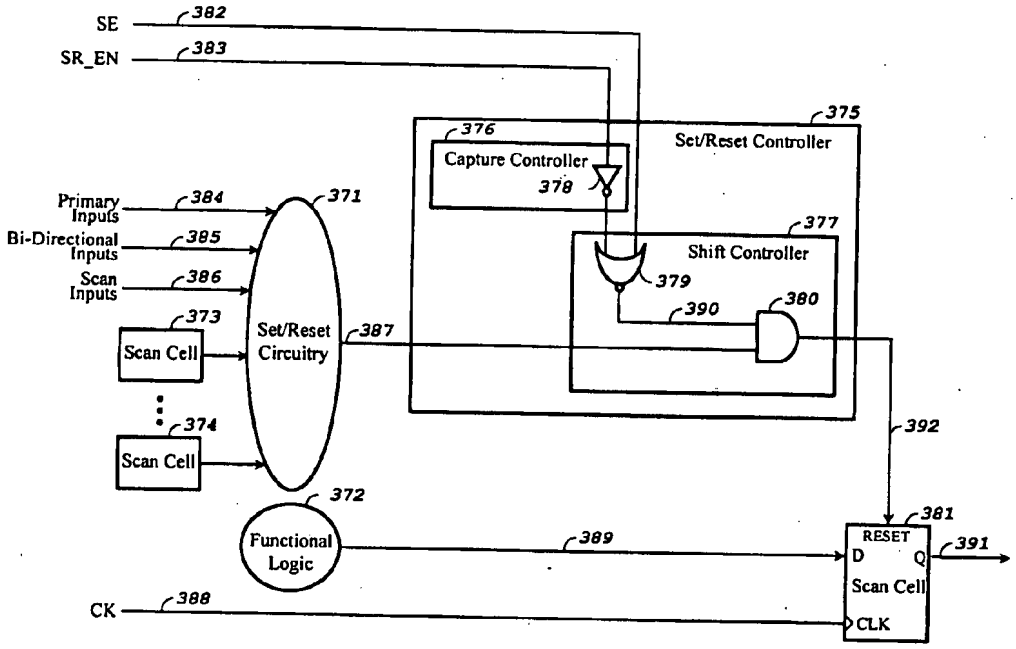

SE
382
SR_EN
383
375
Set/Reset Controller
376
Capture Controller
378
377
Shift Controller
379
380
390
Primary Inputs
384
Bi-Directional Inputs
385
Scan Inputs
386
371
Set/Reset Circuitry
387
373
Scan Cell
374
Scan Cell
372
Functional Logic
389
392
381
RESET
D
Q
391
Scan Cell
CLK
CK
388

FIG. 3C

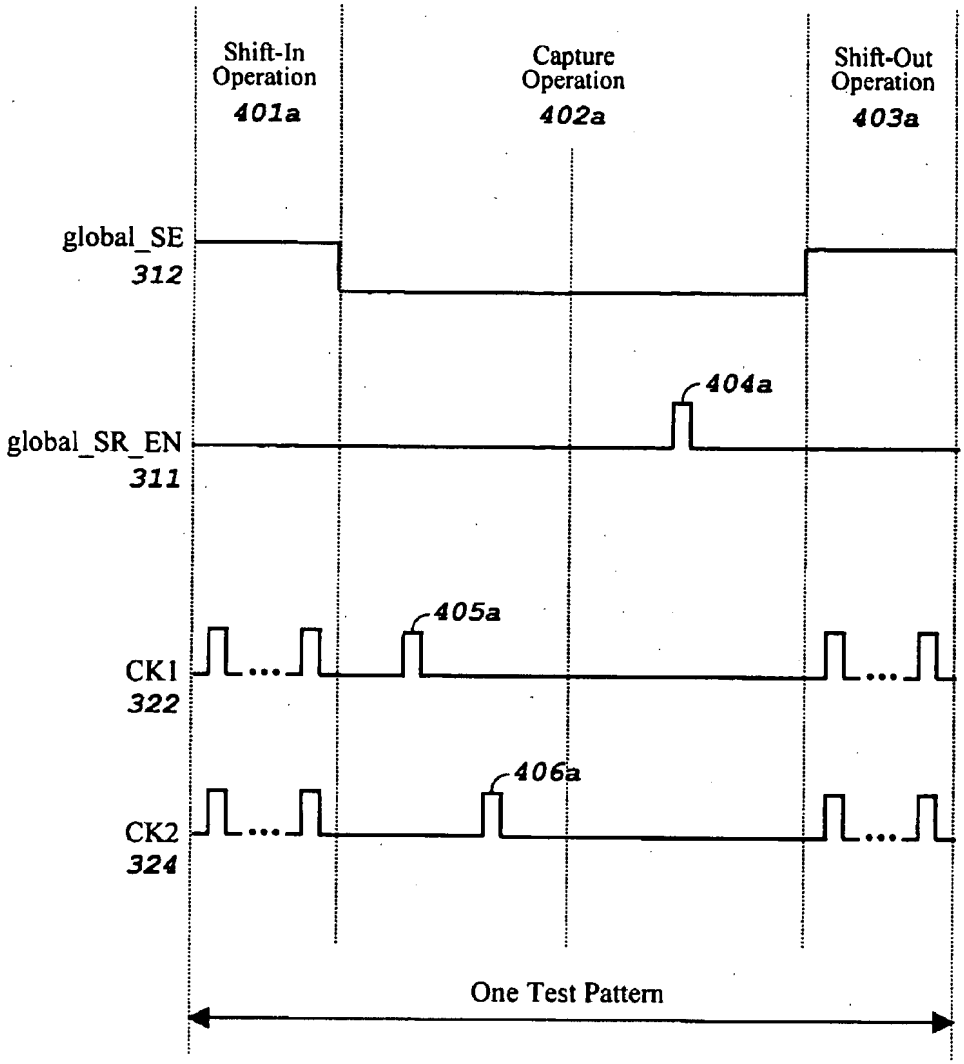

Shift-In Operation
401a
Capture Operation
402a
Shift-Out Operation
403a
global_SE
312
404a
global_SR_EN
311
405a
CK1
322
406a
CK2
324
One Test Pattern

FIG. 4A

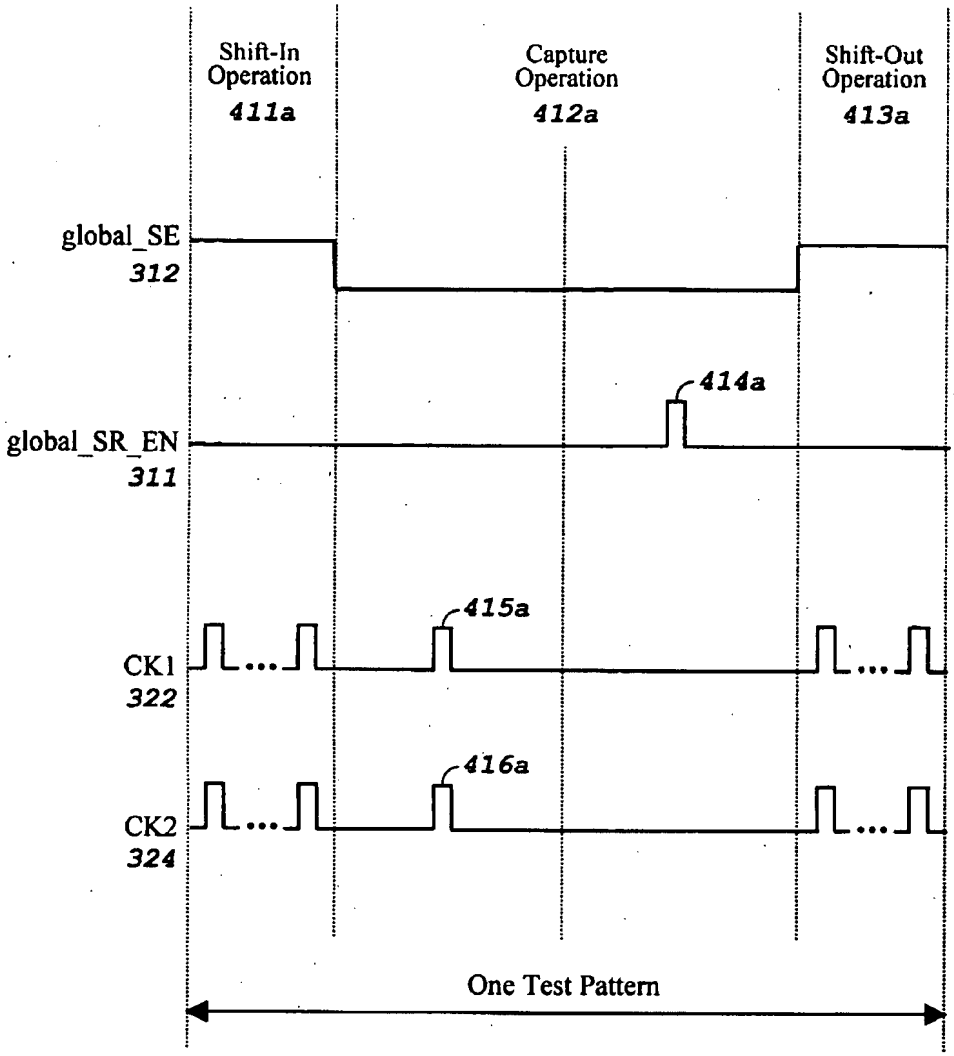

Shift-In Operation
411a
Capture Operation
412a
Shift-Out Operation
413a
global_SE
312
414a
global_SR_EN
311
415a
CK1
322
416a
CK2
324
One Test Pattern

FIG. 4B

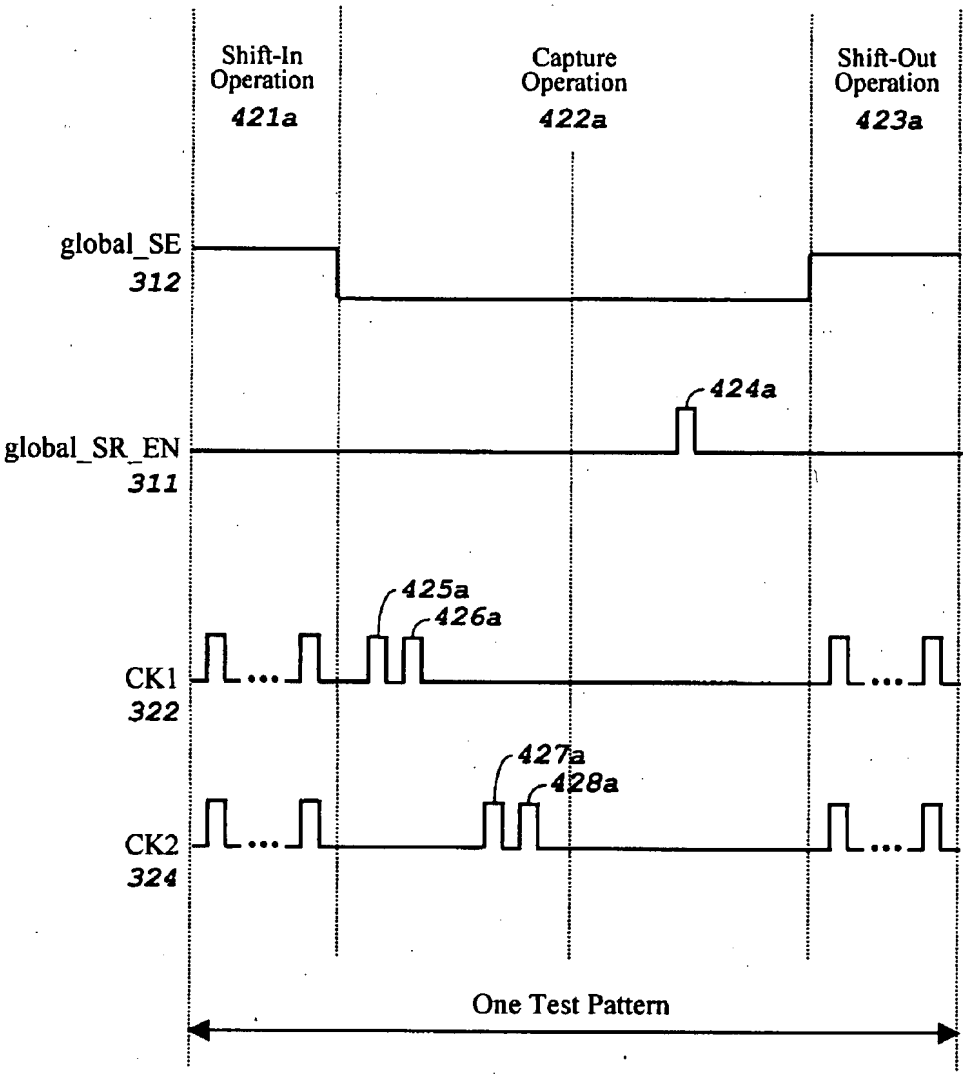

Shift-In Operation
421a
Capture Operation
422a
Shift-Out Operation
423a
global_SE
312
global_SR_EN
311
424a
CK1
322
425a
426a
CK2
324
427a
428a
One Test Pattern

FIG. 4C

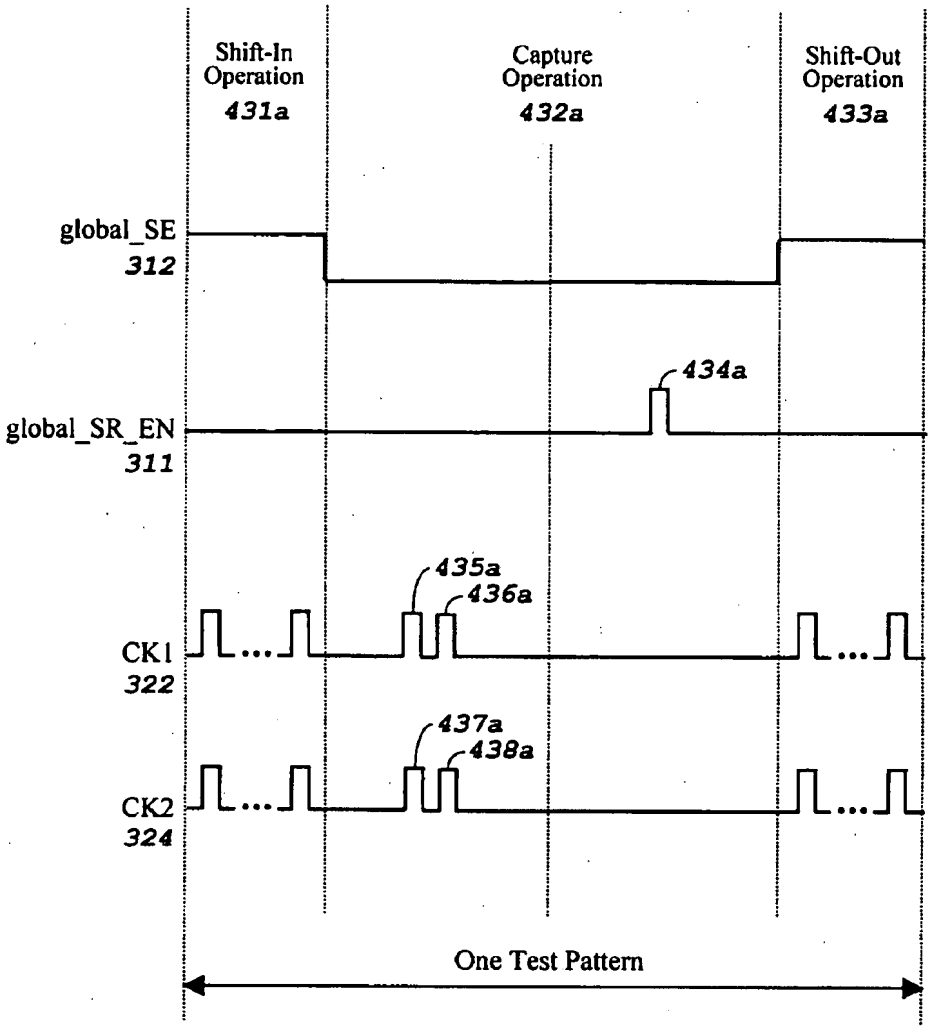

Shift-In Operation
431a
Capture Operation
432a
Shift-Out Operation
433a
global_SE
312
434a
global_SR_EN
311
435a
436a
CK1
322
437a
438a
CK2
324
One Test Pattern

FIG. 4D

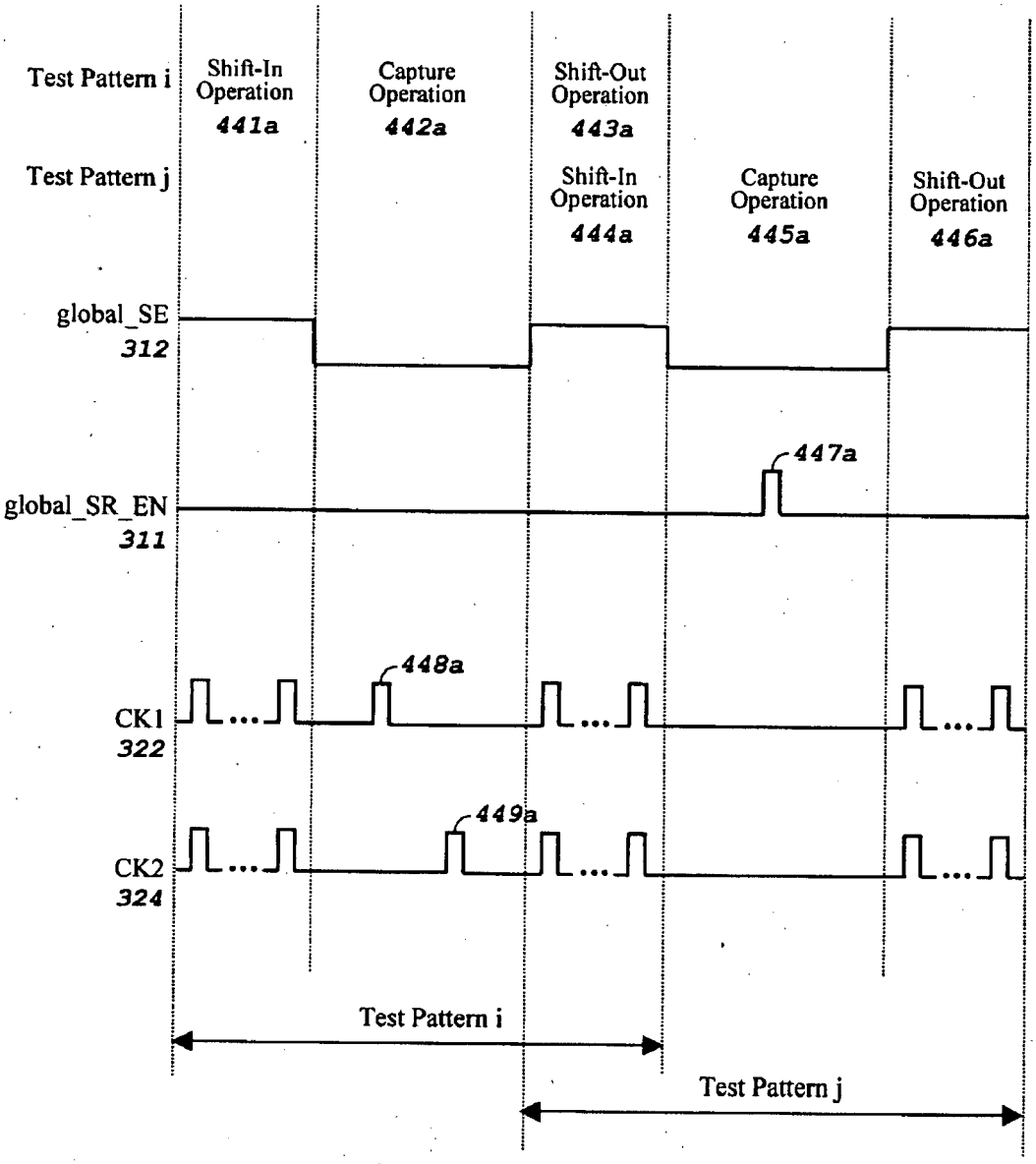

Test Pattern i
Shift-In Operation 441a
Capture Operation 442a
Shift-Out Operation 443a
Test Pattern j
Shift-In Operation 444a
Capture Operation 445a
Shift-Out Operation 446a
global_SE 312
447a
global_SR_EN 311
448a
CK1 322
449a
CK2 324
Test Pattern i
Test Pattern j

FIG. 4E

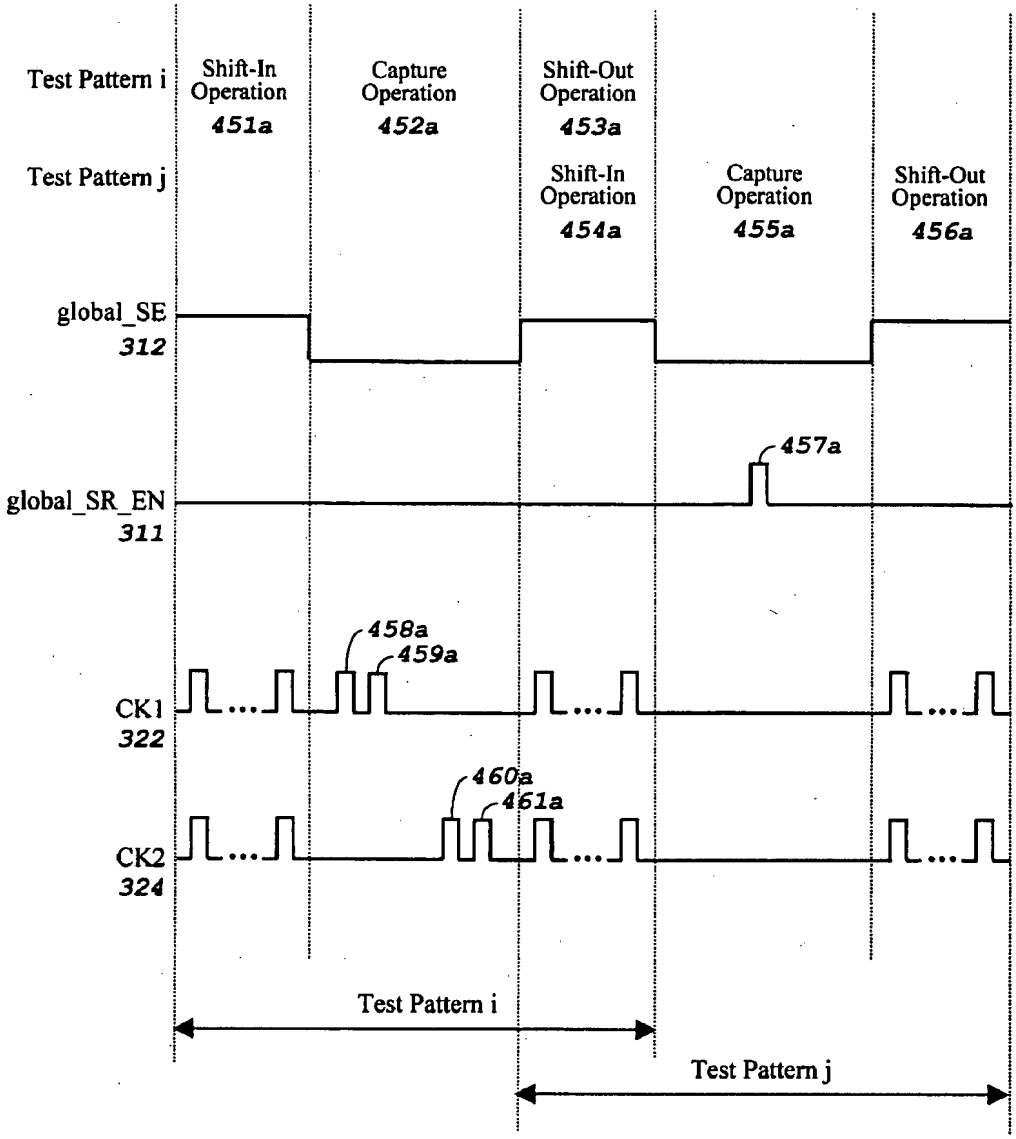

Test Pattern i
Shift-In Operation 451a
Capture Operation 452a
Shift-Out Operation 453a
Test Pattern j
Shift-In Operation 454a
Capture Operation 455a
Shift-Out Operation 456a
global_SE 312
457a
global_SR_EN 311
458a
459a
CK1 322
460a
461a
CK2 324
Test Pattern i
Test Pattern j

FIG. 4F

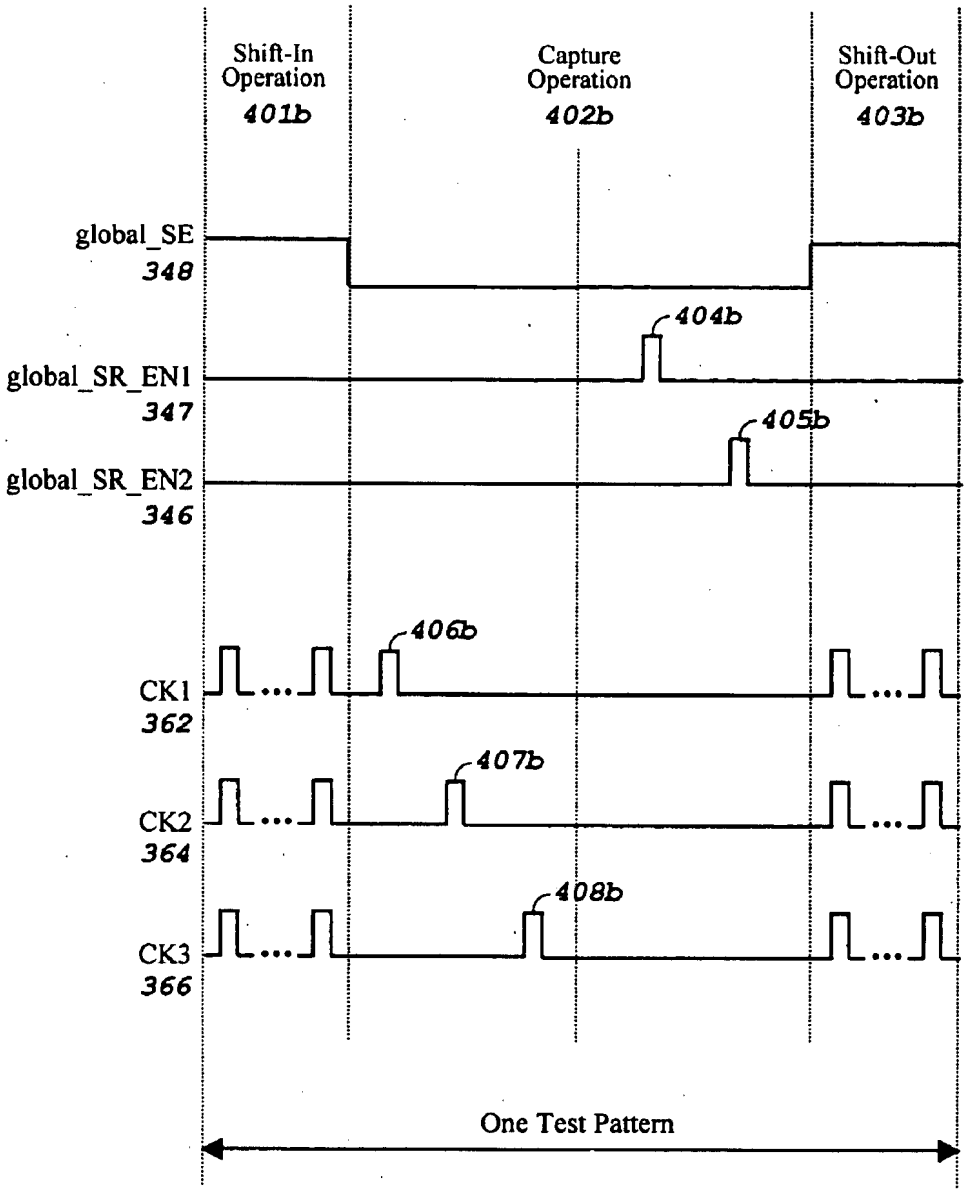

Shift-In Operation
401b
Capture Operation
402b
Shift-Out Operation
403b
global_SE
348
404b
global_SR_EN1
347
405b
global_SR_EN2
346
406b
CK1
362
407b
CK2
364
408b
CK3
366
One Test Pattern

FIG. 4G

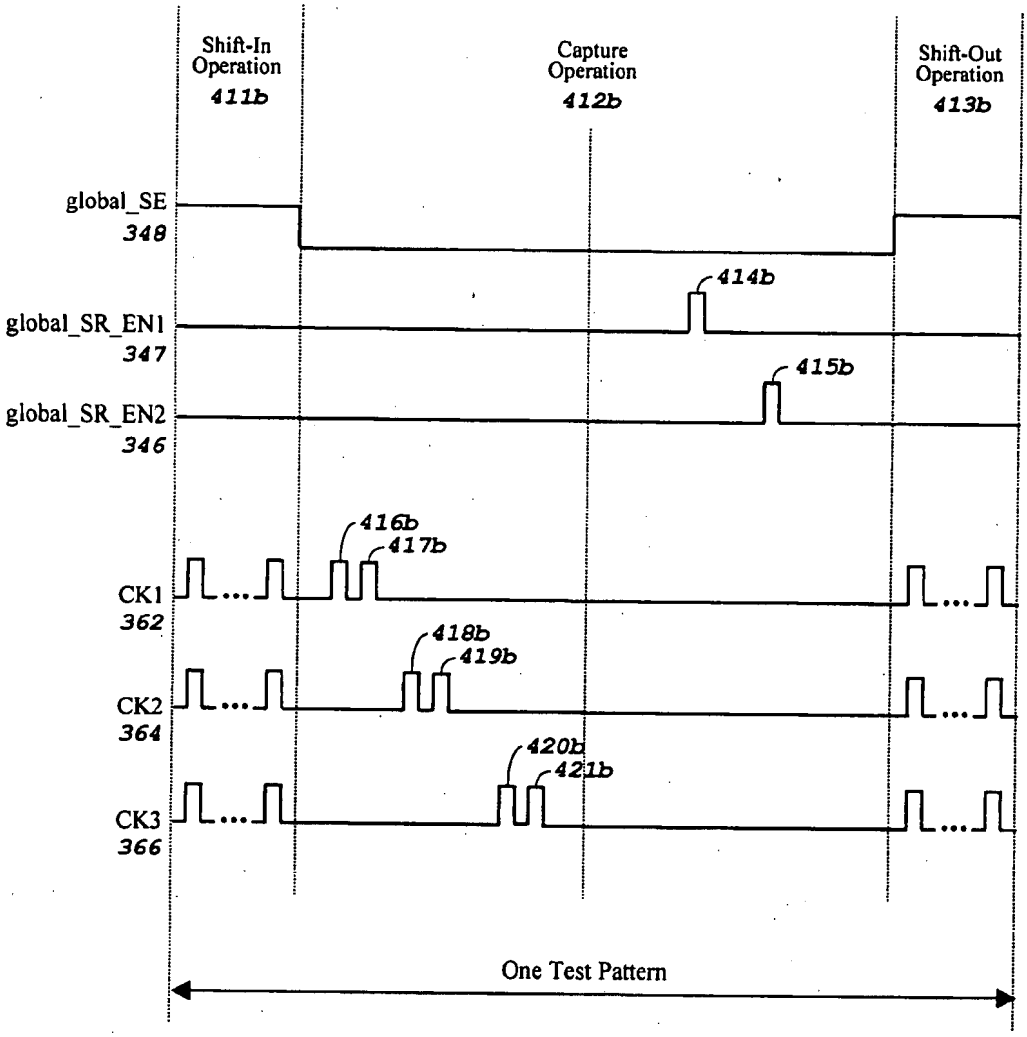

Shift-In Operation
411b
Capture Operation
412b
Shift-Out Operation
413b
global_SE
348
414b
global_SR_EN1
347
415b
global_SR_EN2
346
416b
417b
CK1
362
418b
419b
CK2
364
420b
421b
CK3
366
One Test Pattern

FIG. 4H

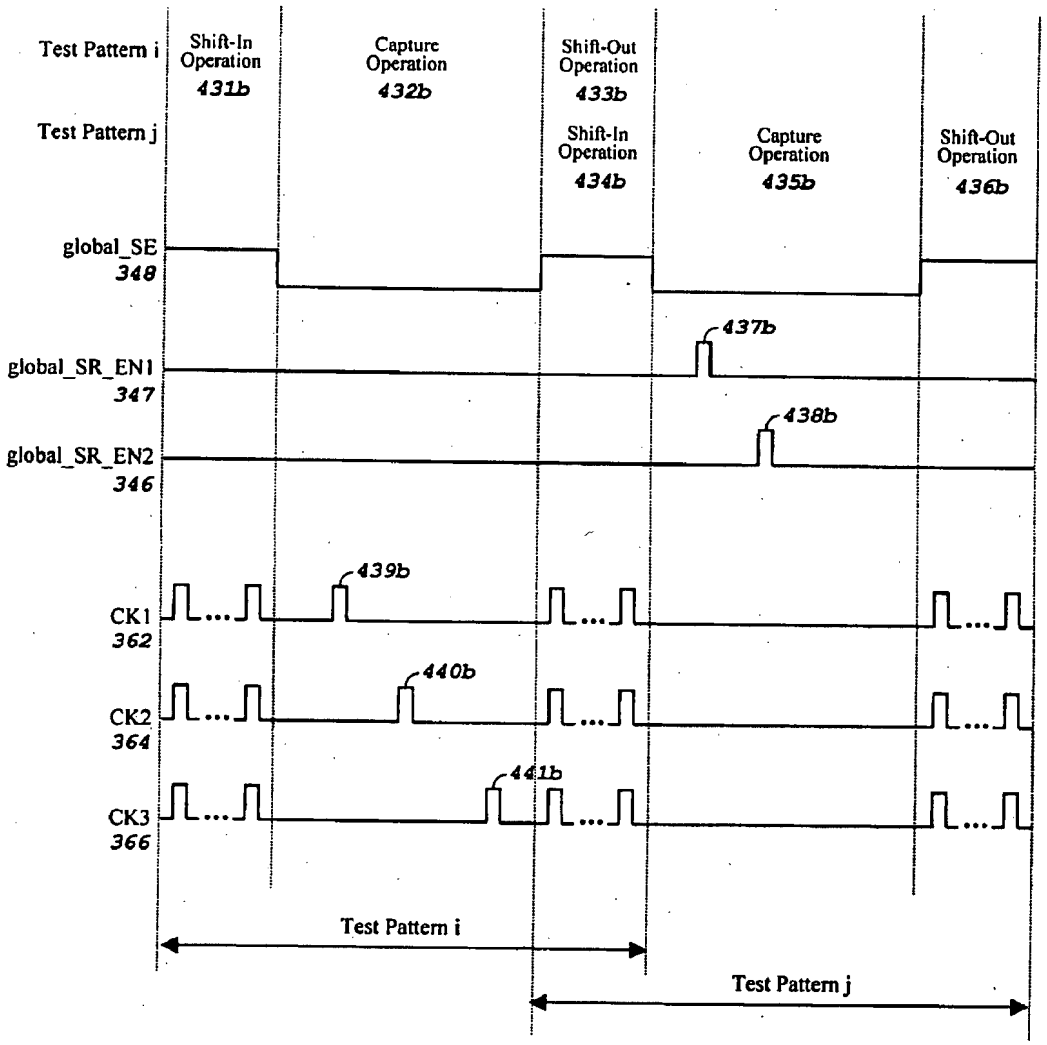

Test Pattern i
Shift-In Operation 431b
Capture Operation 432b
Shift-Out Operation 433b
Test Pattern j
Shift-In Operation 434b
Capture Operation 435b
Shift-Out Operation 436b
global_SE 348
437b
global_SR_EN1 347
438b
global_SR_EN2 346
439b
CK1 362
440b
CK2 364
441b
CK3 366
Test Pattern i
Test Pattern j

FIG. 4I

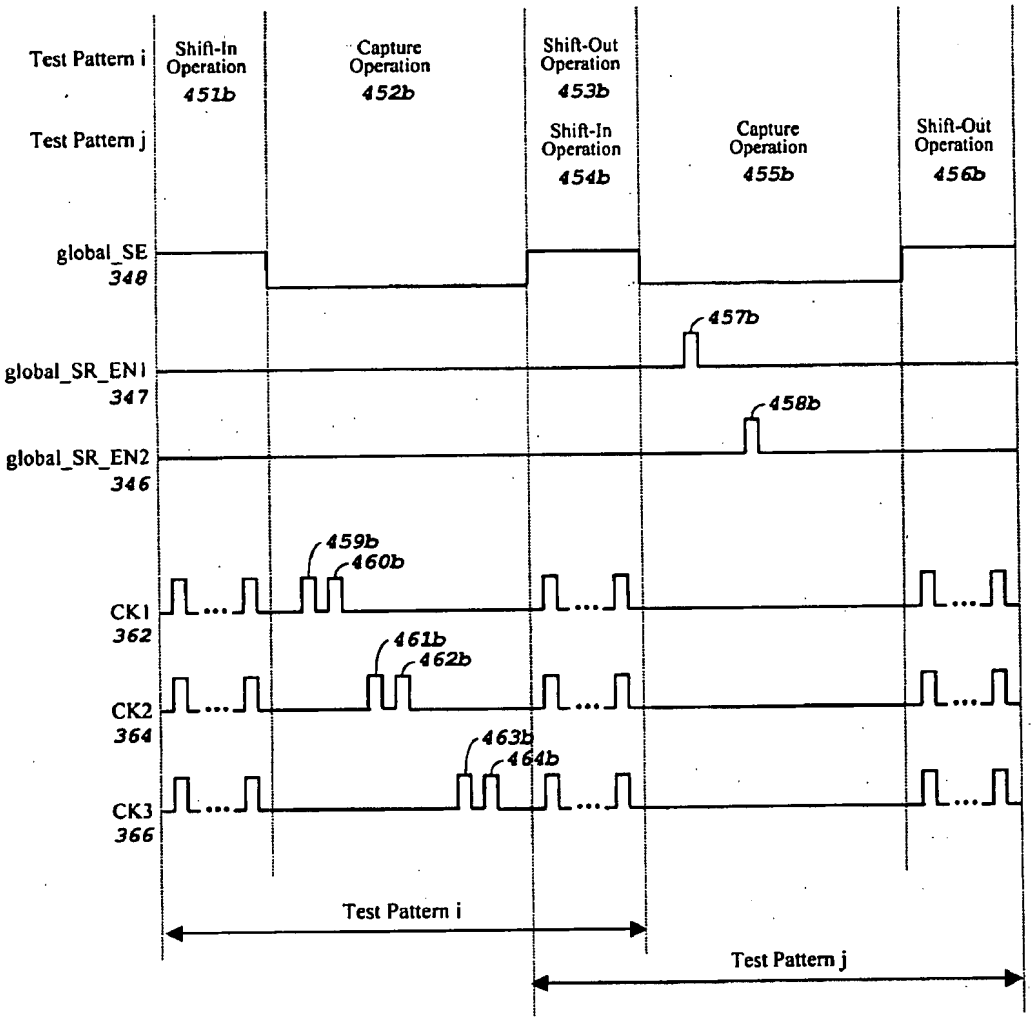

Test Pattern i
Shift-In Operation 451b
Capture Operation 452b
Shift-Out Operation 453b
Test Pattern j
Shift-In Operation 454b
Capture Operation 455b
Shift-Out Operation 456b
global_SE 348
457b
global_SR_EN1 347
458b
global_SR_EN2 346
459b
460b
CK1 362
461b
462b
CK2 364
463b
464b
CK3 366
Test Pattern i
Test Pattern j

FIG. 4J

	Original	Modified
1	input rst, x, d, ck;	input rst, x, d, ck, SE, SR_EN;
2	wire s_rst, c_rst;	wire s_rst, c_rst;
3	reg z, q1, q2;	reg z, q1, q2;
4		
5	assign s_rst = rst & z;	// Added for repair.
6	assign c_rst = rst & x;	wire scan_s_rst;
7	always @(posedge ck)	wire scan_c_rst;
8	z <= x;	
9		// Added to repair sequentially-gated reset.
10	// s_rst is sequentially-gated reset.	assign scan_s_rst = ~(SE | ~SR_EN) & s_rst;
11	always @(posedge ck or posedge s_rst)	// Added to repair combinational gated reset.
12	if (s_rst)	assign scan_c_rst = ~(SE | ~SR_EN) & c_rst;
13	q1 <= 0;	
14	else	assign s_rst = rst & z;
15	q1 <= d;	assign c_rst = rst & x;
16		always @(posedge ck)
17	// c_rst is combinationally-gated reset.	z <= x;
18	always @ (posedge ck or posedge c_rst)	
19	if (c_rst)	// Modified to repair sequentially-gated reset.
20	q2 <= 0;	always @(posedge ck or posedge scan_s_rst)
21	else	if (scan_s_rst)
22	q2 <= d;	q1 <= 0;
23		else
24		q1 <= d;
25		
26		// Modified to repair combinationally-gated reset.
27		always @(posedge ck or posedge scan_c_rst)
28		if (scan_c_rst)
29		q2 <= 0;
30		else
31		q2 <= d;

FIG. 5A

	Original	Modified
1	input x, d, ck;	input x, d, ck, SE, SR_EN;
2	wire d_rst;	wire d_rst;
3	reg g_rst, q1, q2;	reg g_rst, q1, q2;
4		
5	assign d_rst = 1;	// Added for repair.
6	always @(posedge ck)	wire scan_g_rst;
7	g_rst <= x;	wire scan_d_rst;
8		
9	// g_rst is generated reset.	// Added to repair generated reset.
10	always @(posedge ck or posedge g_rst)	assign scan_g_rst = ~(SE | ~SR_EN) & g_rst;
11	if (g_rst)	// Added to repair destructive reset.
12	q1 <= 0;	assign scan_d_rst = ~(SE | ~SR_EN) & d_rst;
13	else	
14	q1 <= d;	assign d_rst = 1;
15		always @(posedge ck)
16	// d_rst is destructive reset.	g_rst <= x;
17	always @ (posedge ck or posedge d_rst)	
18	if (d_rst)	// Modified to repair generated reset.
19	q2 <= 0;	always @(posedge ck or posedge scan_g_rst)
20	else	if (scan_g_rst)
21	q2 <= d;	q1 <= 0;
22		else
23		q1 <= d;
24		
25		// Modified to repair destructive reset.
26		always @(posedge ck or posedge scan_d_rst)
27		if (scan_d_rst)
28		q2 <= 0;
29		else
30		q2 <= d;

FIG. 5B

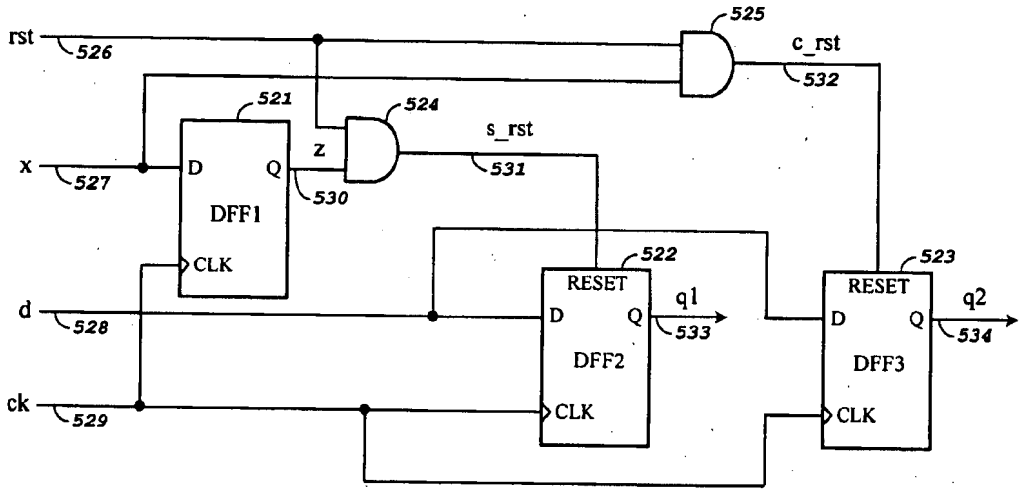

rst
526
525
c_rst
532
521
524
s_rst
531
x
527
D
Q
z
530
DFF1
CLK
522
523
RESET
RESET
d
528
D
Q
q1
533
D
Q
q2
534
DFF2
DFF3
ck
529
CLK
CLK

FIG. 5C

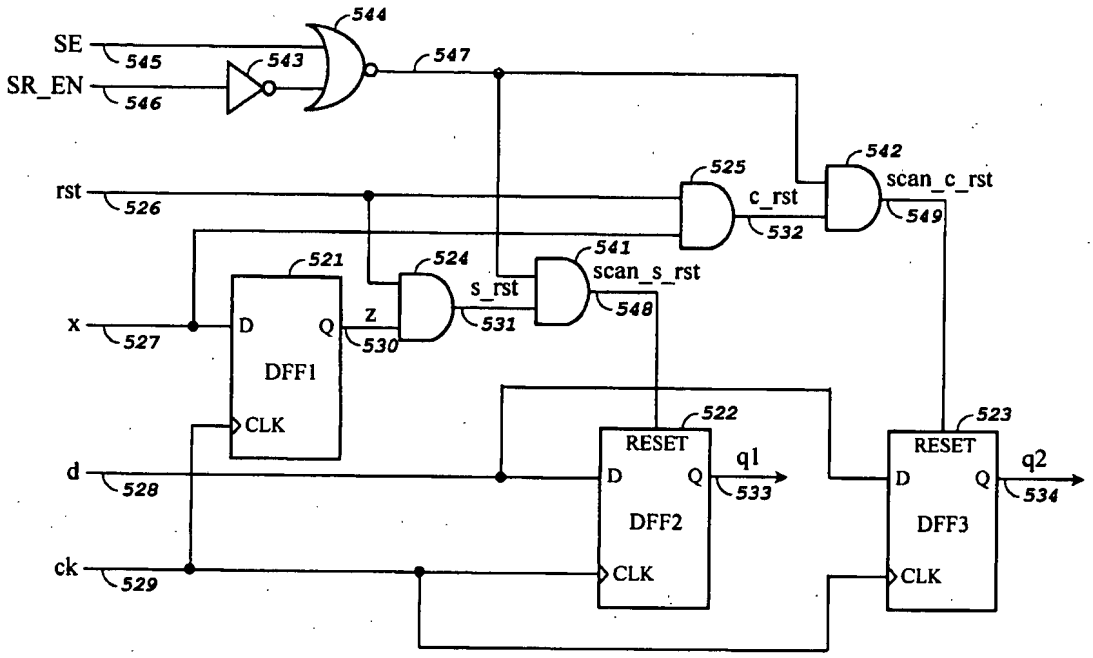

SE
545
SR_EN
546
544
543
547
rst
526
525
c_rst
532
542
scan_c_rst
549
x
527
521
D
Q
DFF1
CLK
z
530
524
s_rst
531
541
scan_s_rst
548
d
528
522
RESET
D
Q
q1
533
DFF2
CLK
523
RESET
D
Q
q2
534
DFF3
CLK
ck
529

FIG. 5D

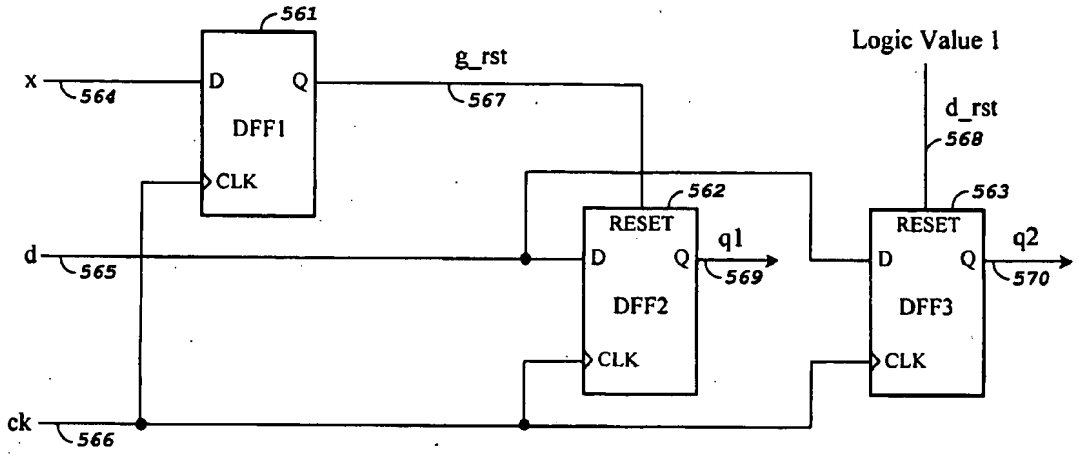

561
x
564
D
Q
DFF1
CLK
g_rst
567
Logic Value 1
d_rst
568
562
563
RESET
q1
RESET
q2
d
565
D
Q
569
D
Q
570
DFF2
DFF3
CLK
CLK
ck
566

FIG. 5E

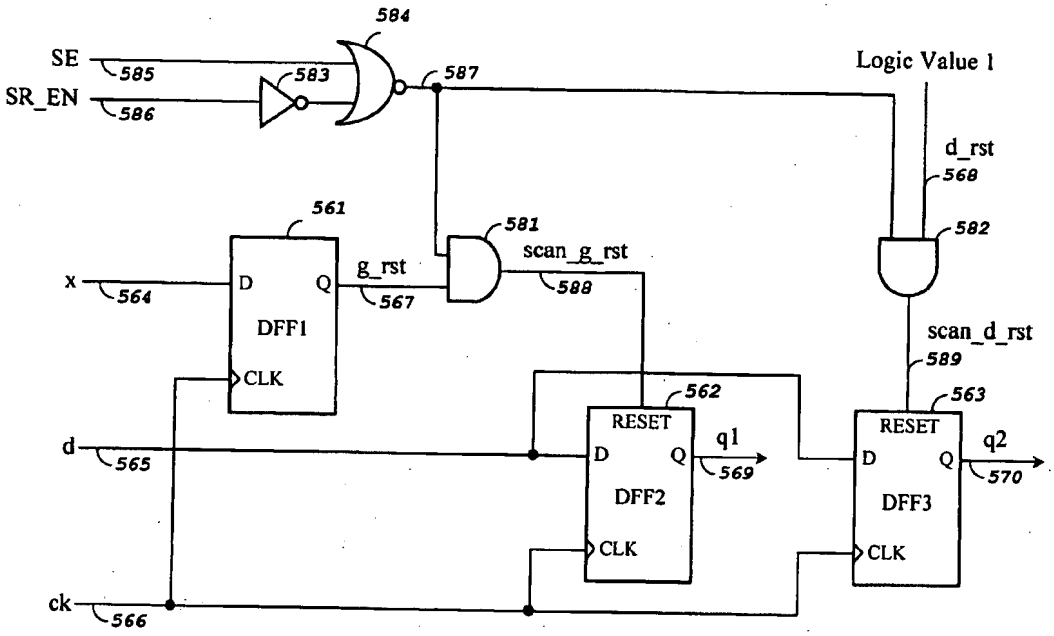

584
SE
585
583
SR_EN
586
587
Logic Value 1
d_rst
568
582
561
581
x
564
D
Q
g_rst
567
scan_g_rst
588
DFF1
CLK
scan_d_rst
589
562
563
RESET
RESET
d
565
D
Q
q1
569
D
Q
q2
570
DFF2
DFF3
CLK
CLK
ck
566

FIG. 5F

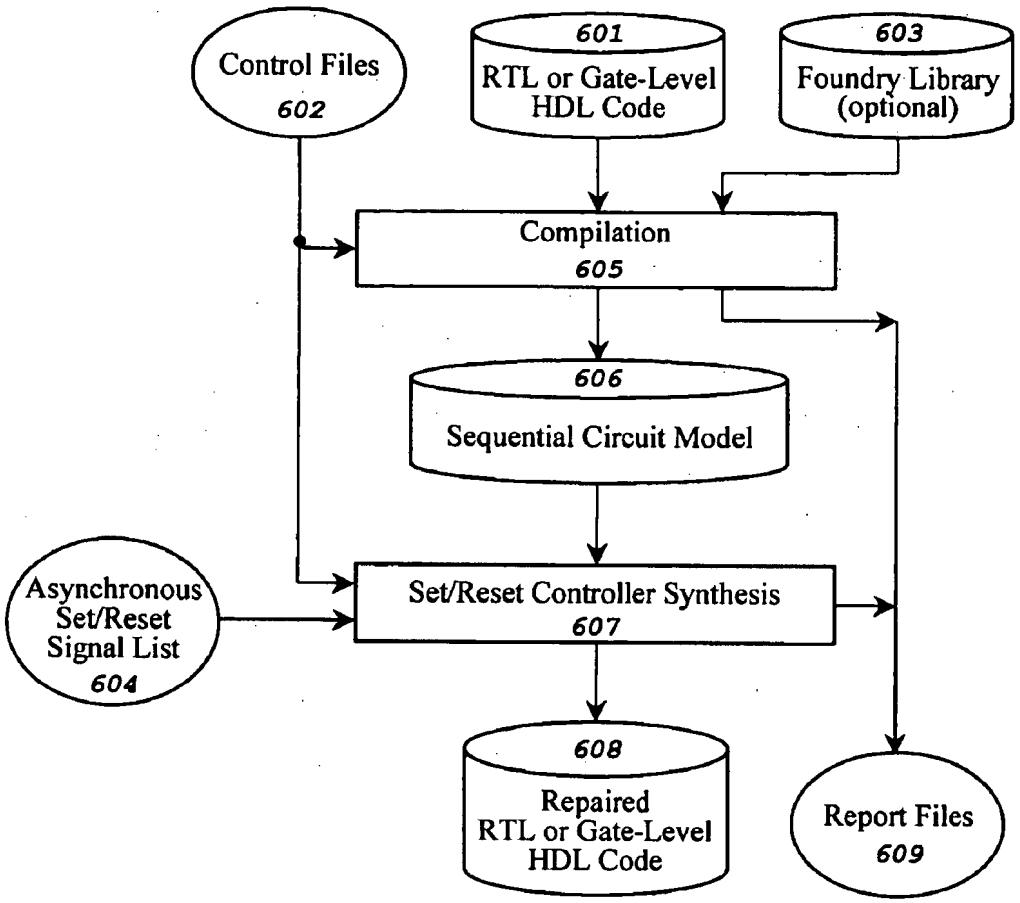

Control Files
602
601
RTL or Gate-Level
HDL Code
603
Foundry Library
(optional)
Compilation
605
606
Sequential Circuit Model
Asynchronous
Set/Reset
Signal List
604
Set/Reset Controller Synthesis
607
608
Repaired
RTL or Gate-Level
HDL Code
Report Files
609

FIG. 6

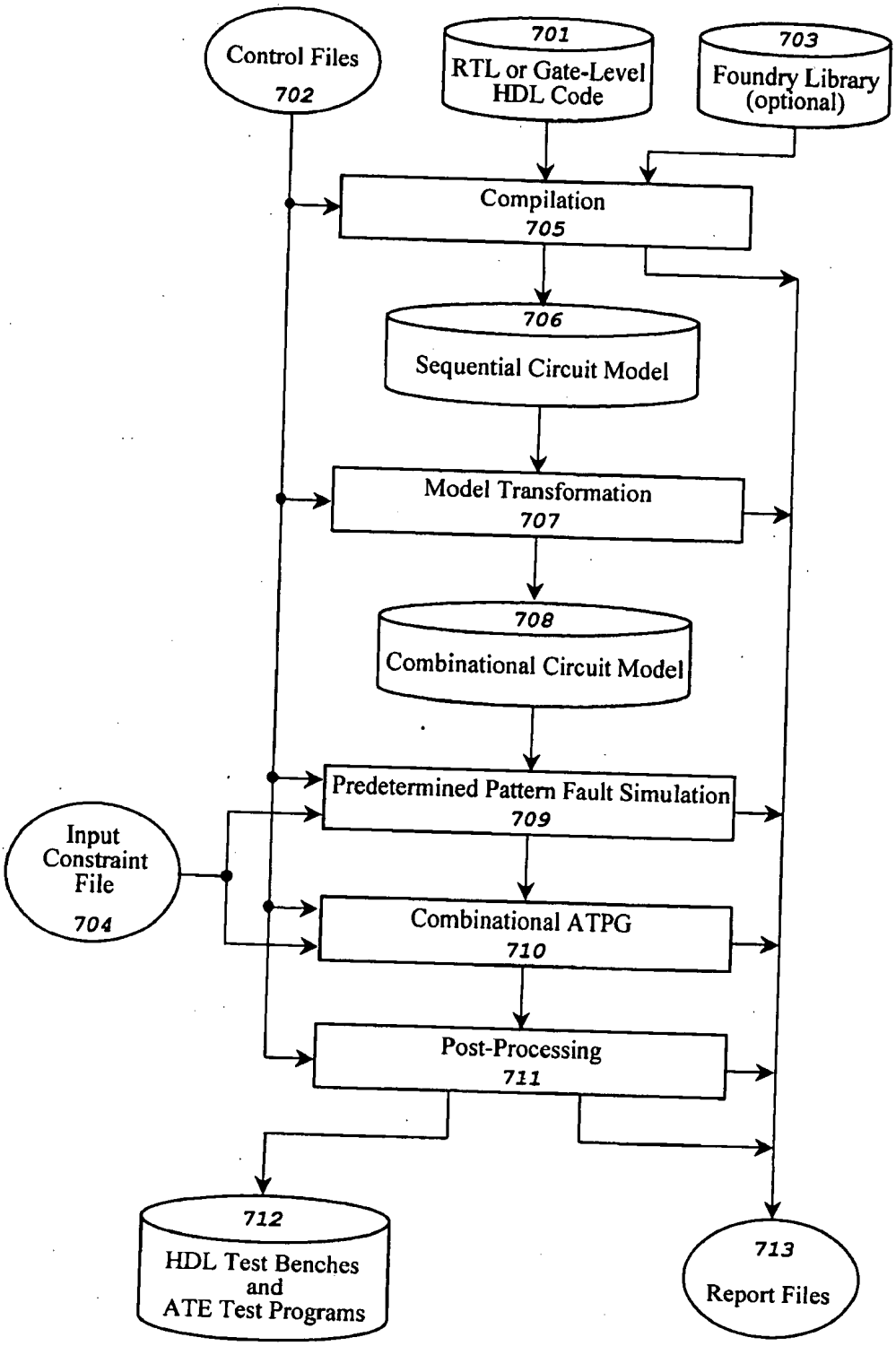

Control Files
702
701
RTL or Gate-Level HDL Code
703
Foundry Library (optional)
Compilation
705
706
Sequential Circuit Model
Model Transformation
707
708
Combinational Circuit Model
Predetermined Pattern Fault Simulation
709
Input Constraint File
704
Combinational ATPG
710
Post-Processing
711
712
HDL Test Benches and ATE Test Programs
713
Report Files

FIG. 7A

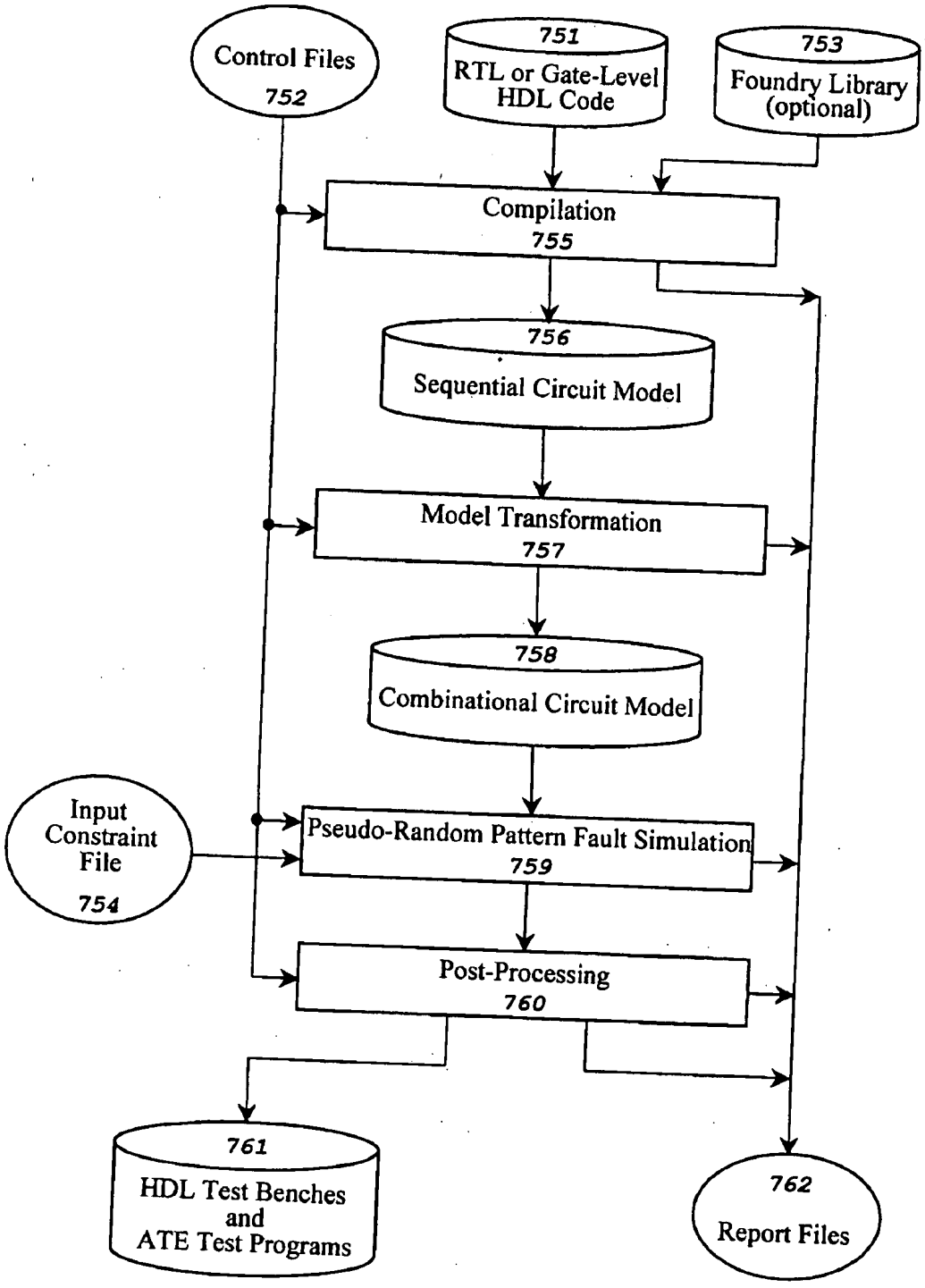

Control Files
752
751
RTL or Gate-Level HDL Code
753
Foundry Library (optional)
Compilation
755
756
Sequential Circuit Model
Model Transformation
757
758
Combinational Circuit Model
Input Constraint File
754
Pseudo-Random Pattern Fault Simulation
759
Post-Processing
760
761
HDL Test Benches and ATE Test Programs
762
Report Files

FIG. 7B

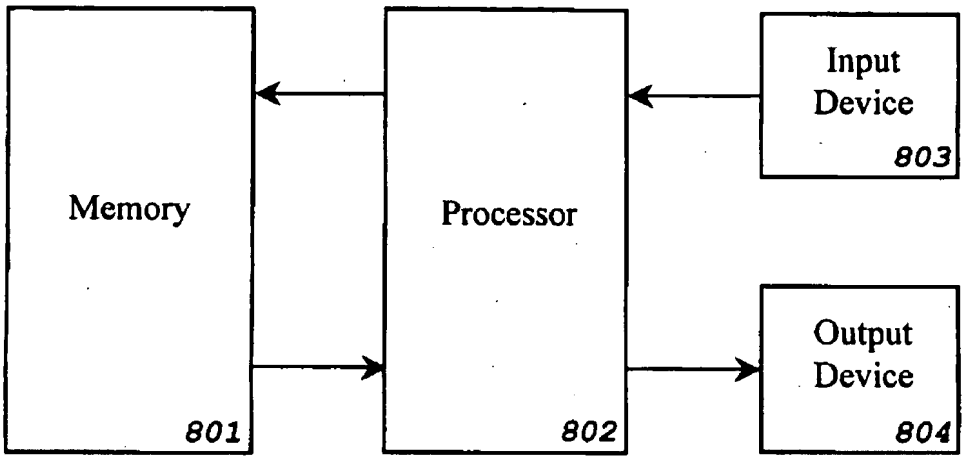

Memory
801
Processor
802
Input Device
803
Output Device
804

FIG. 8

METHOD AND APPARATUS FOR TESTING ASYNCHRONOUS SET/RESET FAULTS IN A SCAN-BASED INTEGRATED CIRCUIT

RELATED APPLICATION DATA

[0001] This application claims the benefit of U.S. Provisional Application No. 60/422,117 filed Oct. 30, 2002, titled "Method and Apparatus for Testing Asynchronous Set/Reset Faults in a Scan-Based Integrated Circuit", which is hereby incorporated by reference.

TECHNICAL FIELD

[0002] The present invention generally relates to the field of logic design and test using design-for-test (DFT) techniques. Specifically, the present invention relates to the field of testing asynchronous set/reset faults in integrated circuits by using scan test techniques.

BACKGROUND

[0003] Design methodologies for complex integrated circuits (IC) have evolved to keep pace with the advances in process technologies. The growing number of transistors that can be integrated onto a single device has resulted in shifting the design process to higher levels of abstraction. Hardware description languages (HDLs) have become widely used for describing the behavior of a circuit at various levels of abstraction. Currently, the most commonly used methodology for integrated circuit design is to use Verilog or VHDL HDL to describe a circuit at the register-transfer level (RTL) and to use computer-aided design (CAD) software called a logic synthesis tool to convert the HDL design description into a functionally-equivalent technology-dependent gate-level netlist, while taking into account user constraints related to timing, power, area, etc. The netlist generated by this synthesis process is later taken through a back-end process in order to create a manufacturable representation of the design.

[0004] Each manufactured integrated circuit must be tested in order to verify its structural correctness. With the ever-increasing scale and complexity of integrated circuits, the goal of achieving high test-quality at a reasonable cost is becoming extremely difficult. Therefore, improving the inherent testability of an integrated circuit is imperative in order to realize this goal.

[0005] Numerous techniques have been developed for improving the testability of an integrated circuit. These techniques are collectively referred to as design-for-test (DFT) techniques.

[0006] Among the various DFT techniques, scan-based design has emerged as the most widely used DFT methodology, encompassing the de-facto scan-test methodology using Scan/ATPG (automatic test pattern generation) as well as the self-test methodology using Logic BIST (built-in self-test).

[0007] In a scan-based integrated circuit, the original memory elements, comprising flip-flops and/or latches, are replaced with scan-equivalent storage elements, called scan cells. These scan cells are allowed to select one of two possible data sources depending on the state of a selected scan enable (SE) signal. When SE is set to logic value 0, the normal data input port is selected. When SE is set to logic value 1, the scan input port is selected. The scan input ports and scan output ports of all scan cells are stitched together in a way so that the scan cells are reconfigured as one or more shift registers called scan chains. These scan chains are either accessed internally during self-test or through external scan input ports and scan output ports during scan-test.

[0008] Three operations are used to test a scan-based integrated circuit. These operations are shift-in, capture and shift-out. During the shift-in operation, the scan enable (SE) signal of all scan cells is set to logic value 1. A stimulus is shifted in through the scan chains to initialize the state of all scan cells present in the integrated circuit. Next, during the capture operation, the scan enable (SE) signal is set to logic value 0. Clocks are applied to all scan cells capturing the circuit's response to the stimulus shifted in by the previous operation through the functional logic. Finally, during the shift-out operation, the scan enable (SE) signal is once again set to logic value 1. The captured test response is shifted out through the scan chains. This test response can be compared directly to a predetermined expected response, or compacted into a signature using a compactor such as a multiple-input signature register (MISR) for later comparison. Typically, the shift-in and shift-out operations occur simultaneously as a single shift operation so that a new stimulus is loaded into the scan chains while the previous captured test response is being shifted out. The test is conducted by repetitively applying a predetermined number of test patterns, each consisting of the simultaneous shift-in/shift-out and capture operations.

[0009] In order for the scan cells to operate as a shift register during the shift-in or shift-out operation, it is necessary to disable the set and reset signals of all scan cells in order to prevent these signals from corrupting the data being shifted in or out through the scan chains. This is easily accomplished in cases where the set and reset signals are controlled externally by forcing these external signals into an inactive state. In situations where this is not the case, a set/reset scan-based DFT design-rule violation is said to exist in the circuit. These set/reset violations, as well as other types of DFT design-rule violations, must be repaired in order to be able to use the scan chains to test a scan-based integrated circuit.

[0010] Repairing DFT design-rule violations in a scan-based integrated circuit involves modifying the design to add additional circuitry and/or external signals that are active only during scan-test or self-test. Current methods for repairing asynchronous set/reset violations can result in race conditions and glitches, or fault coverage loss related to the faults present in the functional circuitry driving the set/reset ports of a scan cell. The following is a summary of the four major prior-art solutions used to fix asynchronous set/reset DFT design-rule violations:

[0011] The first prior-art solution (prior-art #1, **FIG. 2B**) uses a test enable (TE) signal and an external set/reset signal to control the asynchronous set/reset ports of all scan cells for the complete duration of scan-test or self-test. This solution repairs the asynchronous set/reset violations by adding a multiplexor that is controlled by the test enable (TE) signal to select either the original functional asynchronous set/reset path in functional mode, or the external set/reset signal in scan-test or self-test mode. In order to disable the asynchronous set/reset ports during the shift

operation, the external set/reset signal is set to an inactive state allowing the scan chains to operate correctly as a shift register. During the capture operation, the external set/reset signal is toggled to capture data through the asynchronous set/reset ports of the scan cells in order to detect the faults occurring on these ports. Since the functional set/reset logic is never selected during scan-test or self-test, the faults associated with this logic cannot be detected using this scheme. This results in a fault coverage loss that can be significant, depending on the number of faults associated with the functional asynchronous set/reset circuitry present in the circuit, which in turn depends on the size of the set/reset circuitry driving the asynchronous set/reset ports of all scan cells in the circuit.

[0012] The second prior-art solution (prior-art #2, **FIG. 2C**) uses a test enable (TE) signal to disable the asynchronous set/reset ports of all scan cells for the complete duration of scan-test or self-test. This solution repairs the asynchronous set/reset violations by adding an AND gate and an inverter to force all asynchronous set/reset ports into an inactive state using the test enable (TE) signal in scan-test or self-test mode, while allowing the functional set/reset signals to drive the asynchronous set/reset ports in functional mode. While this solution has a lower overhead compared to prior-art #1, it results in greater fault coverage loss since it cannot be used to detect the faults located at the set/reset ports of the scan cells present in the circuit.

[0013] The third prior-art solution (prior-art #3, **FIG. 2D**) uses a scan enable (SE) signal to disable the asynchronous set/reset ports of all scan cells during the shift operation for the complete duration of scan-test or self-test. This solution repairs the asynchronous set/reset violations by adding an AND gate and an inverter to force all asynchronous set/reset ports into an inactive state using the scan enable (SE) signal in scan-test or self-test mode, while allowing the functional set/reset signals to drive the asynchronous set/reset ports during the capture operation as well as during normal operation. This guarantees that the asynchronous set/reset ports of all scan cells are disabled during the shift operation allowing the scan chains to operate correctly as a shift register. The advantage of this solution is that the faults present in the functional circuitry driving the asynchronous set/reset ports of all scan cells can now be propagated and tested during the capture operation resulting in no fault coverage loss as compared to prior-art solutions #1 and #2. In practice however, problems occur when using this solution due to the race condition between the data and set/reset ports that occurs during the capture cycle. This can often result in an unreliable state being captured into the scan cells, followed by pattern mismatches during comparison or compaction, thus invalidating the test.

[0014] The fourth prior-art solution (prior-art #4, **FIG. 2E**) uses an external set/reset enable (ESR_EN) signal to disable the asynchronous set/reset ports of all scan cells during scan-test. This solution repairs the asynchronous set/reset violations by adding a multiplexor gate.

[0015] During the shift operation, the external set/reset enable (ESR_EN) signal is disabled to guarantee that all asynchronous set/reset ports of all scan cells are disabled allowing all scan chains to operate correctly as a shift register. During the capture operation, two options are possible. In one option, ESR_EN is set to allow the functional set/reset signals to drive the asynchronous set/reset ports, while the clocks are disabled, to test the set/reset logic. In the other option, ESR_EN is used to force all asynchronous set/reset ports into an inactive state, while the clocks are used to test the faults on the data ports of the scan cells.

[0016] The advantage of this solution is that the faults present in the functional circuitry driving the asynchronous set/reset ports of all scan cells can now be propagated and tested during the capture operation resulting in no fault coverage loss as compared to prior-art solutions #1 and #2 and in a way that does not create the glitches associated with race conditions between the clock and the set/reset ports of the scan cell. Race conditions, due to ripple reset conditions where setting or resetting a set of scan cells creates an intermediate state forcing additional set of scan cells being set or reset unexpectedly, are solved by using the multiple ripple ESR_EN signals, thus, no glitches are possible.

[0017] However, this solution suffers from two problems. The first problem that the ESR_EN signals must be external pins makes it a difficult solution to implement for pad-limited solutions during scan-test. This might force the designer to choose between implementing this solution with a smaller number of scan chains and longer test time or abandoning this solution to allow for more scan chains. The other problem is with regards to implementing this solution in a self-test environment. Since the ESR_EN signals are not qualified with a scan enable (SE) signal, it is impossible to use this solution in a self-test environment without destroying the contents of the scan chains during shift, hence invalidating the test.

[0018] Therefore, there is a need for an improved asynchronous set/reset DFT design-rule violation repair technique comprising a method, apparatus, and a computer-aided design (CAD) system to ensure correct shift operations, detect asynchronous set/reset faults, and avoid race conditions and glitches that can be used for both scan-test and self-test. In addition, there is a need for a method and a computer-aided design (CAD) system for generating and/or fault simulating test patterns based on the improved technique, in order to test data and set/reset faults in a scan-based integrated circuit.

SUMMARY

[0019] Accordingly, the first primary objective of the present invention is to provide an improved asynchronous set/reset DFT violation repair system to ensure correct shift operations and to detect asynchronous set/reset faults while avoiding race conditions and glitches during scan-test or self-test. This system comprises of a method and apparatus for guaranteeing correct shift operations by disabling the asynchronous set/reset ports of scan cells during the shift operation, while allowing the asynchronous set/reset faults to propagate and to be detected without race conditions and glitches during the capture operation. The present invention further comprises of a computer-aided design (CAD) system for RTL scan synthesis and/or gate-level circuit modification based on this method. The inputs to the CAD system are a set of RTL codes or a gate-level netlist modeled in HDL together with any required scan constraints.

[0020] The present invention uses a global scan enable (SE) signal, one or more global set/reset enable (SR_EN) signals, and some additional logic circuitry to achieve the

stated objective. The scan enable (SE) signal controls the additional logic circuitry to disable the asynchronous set/reset ports of all scan cells during the shift operation. During the capture operation two separate methodologies are possible for testing the asynchronous set/reset faults.

[0021] In the first methodology, two sets of patterns are generated for the capture operation, one set of patterns where the SR_EN signal is permanently set to disable the asynchronous set/reset ports and the clocks are captured in order to test the faults on the data ports of the scan cells, and the other set of patterns where the SR_EN signal is set to enable the set/reset path with no capture clocks being applied, in order to test the faults on the asynchronous set/reset ports of the scan cells.

[0022] In the second methodology, both sets of patterns of the previous methodology are merged to create one set of test patterns where the SR_EN signal acts as a clock that is first disabled while the regular system clocks are applied to capture the faults on the data inputs of the scan cells and later toggled to enable the asynchronous set/reset faults to propagate and to be tested and then disabled in time for the next shift operation. In these two methodologies, since the SR_EN is always disabled when the clocks are being applied, no race conditions of glitches can occur, and since the SR_EN is enabled at some point to allow the asynchronous set/reset faults to propagate, we are guaranteed to be able to thoroughly test the asynchronous set/reset circuitry, hence overcoming all the shortcomings of prior-art solutions #1, #2 and #3.

[0023] Ripple reset glitches where simultaneously setting and/or resetting a set of scan cells causes the circuit to go through intermediate states that generate indeterministic reset glitches on other scan cells are solved by using multiple SR_EN signals to break the ripple reset cycle. Since the SR_EN signals of the present invention can either be generated internally or applied externally this does not result in any additional requirement regarding the number of external pins needed for scan-test. Furthermore since scan enable is used to disable the set/reset ports during the shift operation this solution can easily adapted for either scan-test or self-test, hence overcoming all the shortcomings of prior-art solution #4. The present invention covers the mentioned asynchronous set/reset DFT design-rule violation repair at RTL, gate-level or any other level of abstraction during the design process.

[0024] The second primary objective of the present invention is to provide an improved system for improving fault coverage. This system comprises a method and a computer-aided design (CAD) system for generating and/or fault simulating test patterns to test data faults and set/reset faults in a scan-based integrated circuit, where the asynchronous set/reset violations have been repaired by the asynchronous set/reset violation repair method, in accordance with the present invention.

[0025] The asynchronous set/reset violation repair method and the test pattern generation and/or fault simulation method for a scan-based integrated circuit obtained after such repair, in accordance with the present invention, are summarized as follows:

[0026] (1) Asynchronous Set/Reset Violation Identification

[0027] Generally, the asynchronous set/reset signal of a scan cell is generated by a set/reset circuitry driven by primary inputs, bi-directional primary inputs, scan inputs, and the outputs of scan cells. Its identification as an asynchronous set/reset DFT design-rule violation using testability analysis can be made at RTL, gate-level, or any other level of abstraction during the design process.

[0028] Asynchronous set/reset violations can be classified under four different categories for identification purposes: Sequentially-Gated Set/Reset, Combinationally-Gated Set/Reset, Generated Set/Reset, and Destructive Set/Reset. In a Sequentially-Gated Set/Reset violation, the set/reset signal can be traced back to a specific set/reset source, such as an external set/reset signal, that is gated with the output of a memory element, such as a flip-flop or a latch. In a Combinationally-Gated Set/Reset violation, the set/reset signal can be traced back to a specific set/reset source that is gated with a primary input or the output of a combinational logic block driven by one or more primary inputs. In a Generated Set/Reset violation, the set/reset signal cannot be traced back to a specific set/reset source. In a Destructive Set/Reset violation, the set/reset signal is constantly forced into an active/destructive state by an internal hardwire.

[0029] (2) Asynchronous Set/Reset Violation Repair Circuitry

[0030] (2-1) Set/Reset Controller

[0031] If the asynchronous set/reset signal of a scan cell is identified as an asynchronous set/reset DFT design-rule violation of any of the four types mentioned in (1), the present invention adds a set/reset controller related to the set/reset circuitry and the set/reset ports of the scan cell either automatically or interactively. A set/reset controller is controlled by a scan enable (SE) signal and a set/reset enable (SR_EN) signal. A set/reset controller further comprises of a shift controller and a capture controller.

[0032] (2-2) Shift Controller

[0033] A shift controller comprises circuitry that uses a scan enable (SE) signal to disable the asynchronous set/reset ports of a scan cell, in order to avoid destroying data held by the scan cell during the shift operation. A shift controller can be embedded as part of the set/reset circuitry of a scan cell or placed between the set/reset circuitry and its corresponding scan cell. Furthermore, a scan enable (SE) signal can be generated in an integrated circuit or provided as an external input signal to the device.

[0034] (2-3) Capture Controller

[0035] A capture controller comprises circuitry that uses a set/reset enable (SR_EN) signal to selectively allow the propagation of faults in the set/reset circuitry of a scan cell to the asynchronous set/reset ports of the scan cell during the capture operation. In order for race conditions not to occur, this must be done at a time when all capture clocks are inactive, to avoid the hazardous, simultaneous propagation of signals through the set/reset and data inputs of the scan cells. A capture controller can be embedded as part of the set/reset circuitry of a scan cell or placed between the set/reset circuitry and the corresponding scan cell. Furthermore, a set/reset enable (SR_EN) can be generated in an integrated circuit or provided as an external input signal to the device.

[0036] (3) Asynchronous Set/Reset Violation Repair Operation

[0037] A possible operation of a set/reset controller is as follows: During the shift operation, the scan enable (SE) signal is set to logic value 1, forcing the shift controller to set the asynchronous set/reset ports of all scan cells to the inactive state, preventing the shift in data from being destroyed. Once the shift operation is completed, the circuit enters the capture operation where the scan enable (SE) signal is set to logic value 0. During the first stage of the capture operation, the set/reset enable (SR_EN) signal is set to logic value 0, forcing the asynchronous set/reset ports of all scan cells to remain disabled and the clocks are applied to capture the fault effects propagated to the data ports into the scan cells. During the second stage of the capture operation, all clocks are disabled and the set/reset enable (SR_EN) signal is set to logic value 1, enabling the propagation of the faults in the set/reset circuitry to the scan cells via the asynchronous set/reset ports. In this manner, the asynchronous set/reset faults of a scan cell can be detected without suffering from race conditions or glitchs.

[0038] The following table summarizes a possible implementation of a set/reset controller according to the present invention:

TE	SE	SR_EN	Clock	Mode	Operation
0	X	X	Active	Functional	Normal
1	1	X	Active	Scan-Test or Self-Test	Shift
1	0	0	Active	Scan-Test or Self-Test	Capture (Data Faults)
1	0	1	Inactive	Scan-Test or Self-Test	Capture (Set/Reset Faults)

[0039] (4) Test Pattern Generation for Data and Set/Reset Faults

[0040] Once all asynchronous set/reset violations in a scan-based integrated circuit are repaired, test pattern generation and/or fault simulation is performed on the repaired circuit in order to improve the fault coverage for set/reset as well as data faults. This method comprises the following computer-implemented steps:

[0041] (4-1) Compile the HDL (Hardware Description Language) Code Modeled at RTL (Register-transfer Level) or Gate-level That Represents the Repaird Scan-based Integrated Circuit into a Sequential Circuit Model.

[0042] (4-2) Specify Input Constraints on Clocks, the Set/Reset Enable (SR_EN) Signal, and the Scan Enable (SE) Signal of the Repaird Scan-based Integrated Circuit.

[0043] (4-3) Transform the Sequential Circuit Model into an Equivalent Combinational Circuit Model.

[0044] (4-4) Generate and/or Fault Simulate Test Patterns According to the Specified Input Constraints and the Combinational Circuit Model.

[0045] In summary, the present invention provides an improved asynchronous set/reset violation repair technique, comprising a method, apparatus, and a computer-aided design (CAD) system, to ensure correct shift operations and detect asynchronous set/reset faults while avoiding race conditions and glitches. In addition, the present invention provides a method and a computer-aided design (CAD) system for generating and/or fault simulating test patterns to test data and set/reset faults in a scan-based integrated circuit, where asynchronous set/reset DFT design-rule violations are repaired according to the present invention. As a result, all faults in the set/reset circuitry are detected using test patterns that are free of all race conditions and glitches, and a higher fault coverage is achieved.

THE BRIEF DESCRIPTION OF DRAWINGS

[0046] The above and other objects, advantages and features of the invention will become more apparent when considered with the following specification and accompanying drawings wherein:

[0047] **FIG. 1A** shows an example integrated circuit design before scan synthesis is performed;

[0048] **FIG. 1B** shows the resulting design after scan synthesis is performed on the design shown in **FIG. 1A**;

[0049] **FIG. 2A** shows an example design with an asynchronous reset violation;

[0050] **FIG. 2B** shows the result of applying the prior-art #1 solution to repair the asynchronous reset violation shown in **FIG. 2A**;

[0051] **FIG. 2C** shows the result of applying the prior-art #2 solution to repair the asynchronous reset violation shown in **FIG. 2A**;

[0052] **FIG. 2D** shows the result of applying the prior-art #3 solution to repair the asynchronous reset violation shown in **FIG. 2A**;

[0053] **FIG. 2E** shows the result of applying the prior-art #4 solution to repair the asynchronous reset violation shown in **FIG. 2A**;

[0054] **FIG. 3A** shows a block diagram of two set/reset controllers in a design without any ripple structure, in accordance with the present invention;

[0055] **FIG. 3B** shows a block diagram of three set/reset controllers in a design with a two-stage ripple structure, in accordance with the present invention;

[0056] **FIG. 3C** shows an embodiment of a set/reset controller, in accordance with the present invention;

[0057] **FIG. 4A** shows a timing diagram for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with non-overlapping single-capture clocks;

[0058] **FIG. 4B** shows a timing diagram for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with overlapping single-capture clocks;

[0059] **FIG. 4C** shows a timing diagram for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where both data

faults and set/reset faults are detected -during the same capture operation with non-overlapping at-speed double-capture clocks;

[0060] **FIG. 4D** shows a timing diagram for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with overlapping at-speed double-capture clocks;

[0061] **FIG. 4E** shows a timing diagram for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where data faults and set/reset faults are detected during two capture operations with non-overlapping single-capture clocks;

[0062] **FIG. 4F** shows a timing diagram for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where data faults and set/reset faults are detected during two capture operations with non-overlapping at-speed double-capture clocks;

[0063] **FIG. 4G** shows a timing diagram for testing the design with a two-stage ripple structure shown in **FIG. 3B**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with non-overlapping single-capture clocks;

[0064] **FIG. 4H** shows a timing diagram for testing the design with a two-stage ripple structure shown in **FIG. 3B**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with non-overlapping at-speed double-capture clocks;

[0065] **FIG. 4I** shows a timing diagram for testing the design with a two-stage ripple structure shown in **FIG. 3B**, in accordance with the present invention, where both data faults and set/reset faults are detected during two capture operations with non-overlapping single-capture clocks;

[0066] **FIG. 4J** shows a timing diagram for testing the design with a two-stage ripple structure shown in **FIG. 3B**, in accordance with the present invention, where both data faults and set/reset faults are detected during two capture operations with non-overlapping at-speed double-capture clocks;

[0067] **FIG. 5A** shows an example set of RTL (register-transfer level) Verilog codes before and after a sequentially-gated reset violation and a combinationally-gated reset violation are repaired, in accordance with the present invention;

[0068] **FIG. 5B** shows an example set of RTL (register-transfer level) Verilog codes before and after a generated reset violation and a destructive reset violation are repaired, in accordance with the present invention;

[0069] **FIG. 5C** shows the gate-level circuit model corresponding to the original RTL (register-transfer level) code shown in **FIG. 5A**;

[0070] **FIG. 5D** shows the gate-level circuit model obtained after the sequentially-gated reset violation and the combinationally-gated reset violation shown in **FIG. 5C** are repaired, in accordance with the present invention;

[0071] **FIG. 5E** shows the gate-level circuit model corresponding to the original RTL (register-transfer level) code shown in **FIG. 5B**;

[0072] **FIG. 5F** shows the gate-level circuit model after the generated reset violation and the destructive reset violation shown in **FIG. 5E** are repaired, in accordance with the present invention;

[0073] **FIG. 6** shows a flow diagram of the method for repairing asynchronous set/reset violations at either RTL (register-transfer level) or gate-level, in accordance with the present invention;

[0074] **FIG. 7A** shows a flow diagram of the method for generating test patterns for data faults and set/reset faults in scan-test mode, in accordance with the present invention;

[0075] **FIG. 7B** shows a flow diagram of the method for generating test patterns for data faults and set/reset faults in self-test mode, in accordance with the present invention; and

[0076] **FIG. 8** shows an example electronic design automation system in which the method for repairing asynchronous set/reset violations at either RTL (register-transfer level) or gate-level and the method of generating test patterns for data faults and set/reset faults, in accordance with the present invention, may be implemented.

DETAILED DESCRIPTION OF THE INVENTION

[0077] The following description is presently contemplated as the best mode of carrying out the present invention. This description is not to be taken in a limiting sense but is made merely for the purpose of describing the principles of the invention. The scope of the invention should be determined by referring to the appended claims.

[0078] **FIG. 1A** shows an example integrated circuit design **136** before scan synthesis is performed. The design **136** has four clock domains CD**1101** to CD**4104**, three crossing clock-domain logic blocks CCD**1105** to CCD**3107**, primary inputs **108** to **111**, primary outputs **116** to **119**, and bi-directional pins **120** to **123**. In addition, it has four system clocks CK**1112** to CK**4115**. Furthermore, memory elements ME exist in four clock domains CD**1101** to CD**4104**.

[0079] **FIG. 1B** shows the resulting design **167** after scan synthesis is performed on the design **136** shown in **FIG. 1A**. After scan synthesis is performed, all or part of original memory elements ME are replaced with scan cells SC. In addition, the scan cells SC are stitched into one or more scan chains SCN, which can be accessed by scan inputs **159** to **162** and scan outputs **163** to **166**. Note that a scan cell can be a multiplexed-type D flip-flop, a two-port D flip-flop, or a LSSD (level-sensitive scan design) SRL (shift register latch). A scan cell can accept an input value either from its data input port connected to a functional logic block or its scan input port connected to the output of another scan cell or an external scan input, depending on the value of its corresponding scan enable (SE) signal. When a scan enable (SE) signal is enabled, usually with logic value 1, any scan cell under its control accepts its input value from its scan input port. Generally, scan enable signals SE**1155** to SE**4158**, together with test enable signals TE**1151** to TE**4154**, are also used to repair various DFT (design-for-test) design rule violations, including asynchronous set/reset

violations. In addition, test enable signals TE1151 to TE4154 can be driven by a test mode selection signal, say TE, during scan-test or self-test.

[0080] A scan-based integrated circuit, such as the one shown in FIG. 1B, can be tested in either scan-test mode or self-test mode, by repeating three operations: shift-in, capture, and shift-out, until a limiting criteria is reached. The three operations are described bellow:

[0081] During the shift-in operation, a stimulus is shifted through scan inputs 159 to 162 into all scan cells SC in all scan chains SCN within the four clock domains CD1101 to CD4104, simultaneously. The stimulus is either a predetermined stimulus supplied from an ATE (automatic test equipment) in scan-test mode or a pseudo-random stimulus automatically generated in the scan-based integrated circuit using a pseudo-random pattern generator (PRPG) in self-test mode. After the shift-in operation is completed, capture clocks CK1112 to CK4115 are applied to all clock domains, CD1101 to CD4104, to capture the test response into scan cells SC. After the capture operation is completed, the test responses held by all scan cells are shifted out through scan outputs 163 to 166 during the shift-out operation while the next stimulus is shifted into all scan cells SC at the same time. The shifted-out test response is either compared directly with the expected response on an ATE in scan-test mode or compacted by a compactor, such as a multiple-input signature register (MISR), in self-test mode.

[0082] In any scan-based DFT (design-for-test) technique, the asynchronous set/reset ports of all scan cells must be disabled during the shift operation, including shift-in and shift-out; otherwise, the data that are being shifted into scan chains may be destroyed. If an asynchronous set/reset signal is not controlled directly by a primary input during scan-test or a BIST (built-in self-test) controller during self-test, it will be difficult or even impossible to disable the asynchronous set/reset signal during the shift operation. This is a scan-based DFT design rule violation that must be repaired.

[0083] Generally, there are four types of asynchronous set/reset violations: sequentially-gated set/reset violations, combinationally-gated set/reset violations, generated set/reset violations, and destructive set/reset violations. In a sequentially-gated set/reset violation, the set/reset signal of a scan cell can be traced back to a specified set/reset source gated with the output of a memory element such as a flip-flop or a latch. In a combinationally-gated set/reset violation, the set/reset signal of a scan cell can be traced back to a specified set/reset source gated with a primary input or the output of a combinational logic block. In a generated set/reset violation, the set/reset signal of a scan cell cannot be traced back to any primary input specified as a set/reset source. In a destructive set/reset violation of a scan cell, the set/reset signal is stuck at a certain logic value that sets or resets the scan cell constantly.

[0084] FIG. 2A shows an example design 200 with an asynchronous reset violation. The asynchronous reset signal 210 of the scan cell 205 violates the asynchronous set/reset DFT design rule since it is not controlled directly by a primary input. The asynchronous reset signal 210 is generated by a set/reset circuitry 203, driven by primary inputs 206, bi-directional primary inputs 207, external scan inputs 208, and the outputs of scan cells 201, 202, etc. During the shift operation, one must disable the asynchronous reset signal 210 by forcing logic value 0 on the signal. This puts strong constraints on the values that can be shifted into scan cells 201, 202, etc., as well as the values that primary inputs 206, bi-directional primary inputs 207, and scan inputs 208 can hold during the shift operation. In scan-test based ATPG (automatic test pattern generation), these constraints can result in long test patterns (comprising stimuli and test responses) and low fault coverage. In a self-test based environment, not satisfying these constraints will cause mismatches during compaction, thus invalidating the test.

[0085] FIG. 2B shows the result 220 of applying the prior-art #1 solution to repair the asynchronous reset violation shown in FIG. 2A. This solution uses a multiplexor 221 controlled by the test enable (TE) signal 222 to select either the original asynchronous set/reset signal 210 or an external reset signal RST 223 to provide a reset signal to the scan cell 205. During the entire test process, the external reset signal RST 223 is selected. As a result, the reset port of the scan cell 205 is disabled and the shift operation can be conducted correctly. In addition, the external reset signal RST 223 toggles during the capture operation. As a result, all faults propagating from the external reset signal RST 223 to the reset port of the scan cell 205 through the multiplexor 221 could be detected. However, asynchronous set/reset faults present in the set/reset circuitry 203 can never be detected. This may result in significant fault coverage loss when there are many asynchronous set/reset faults in the asynchronous set/reset circuitry 203.

[0086] FIG. 2C shows the result 240 of applying the prior-art #2 solution to repair the asynchronous reset violation shown in FIG. 2A. One inverter 241 and one AND gate 242 are used instead of the multiplexor 221 used in FIG. 2B. This prior-art solution does not need any external set/reset signal, such as RST 223 shown in FIG. 2B. This solution has lower overhead but yields more fault coverage loss than prior-art #1, as it cannot detect any faults present at the set/reset ports of scan cells.

[0087] FIG. 2D shows the result 260 of applying the prior-art #3 solution to repair the asynchronous reset violation shown in FIG. 2A. This solution uses a scan enable (SE) signal 263 together with an AND gate 262 and an inverter 261 to disable the asynchronous reset port of the scan cell 205. This solution ensures that the asynchronous reset port of the scan cell 205 is disabled during the shift operation. In addition, the asynchronous set/reset faults in the set/reset circuitry 203 can be propagated to the scan cell 205 during the capture operation. Thus, unlike the prior-art #1 and prior-art #2 solutions, there will be no fault coverage loss theoretically. The problem with this solution is that any value change at the data port and asynchronous reset port of the scan cell 205 can occur and be captured simultaneously, when the clock CK 209 is applied. As a result, race conditions and glitches may occur on the Q output 212 of the scan cell 205 during the capture operation. This will cause pattern mismatches during comparison or compaction, thus invalidating the test.

[0088] FIG. 2E shows the result 280 of applying the prior-art #4 solution to repair the asynchronous reset violation shown in FIG. 2A. This solution uses a multiplexor 281 controlled by the external set/reset enable (ESR_EN) signal 282 to disable the asynchronous reset port of the scan cell 205 during scan-test. During the shift operation, the

ESR_EN signal **282** is set to logic value 1 so that any data being shifted into the scan cell **205** will not be destroyed. During the capture operation, two options are possible. In one option, the ESR_EN signal **282** is set to logic value 0 to allow faults in the set/reset circuitry **203** to be detected. In the other option, the ESR_EN signal **282** is set to logic value 1 to disable the asynchronous reset port of the scan cell **205** while the clock CK **209** is applied to test faults propagated to the data port **211** of the scan cell **205**. In addition, being able to disable the asynchronous reset port of the scan cell **205** also helps to prevent any glitch at the output **210** of the set/reset circuitry **203** from affecting the state of the scan cell **205**.

[**0089**] The advantage of this solution is that the faults in the set/reset circuitry **203** can now be propagated and tested during the capture operation and no glitches will be caused due to race conditions between the clock CK **209** and the asynchronous reset port of the scan cell **205**. In addition, by properly controlling multiple ESR_EN signals, one can avoid any glitches due to a ripple set/reset condition where setting or resetting a set of scan cells creates an intermediate state forcing another set of scan cells to be set or reset unexpectedly.

[**0090**] However, this solution suffers from two problems: First, the ESR_EN signal needs to be an external pin, making it infeasible for a design with a tight pin count budget. Second, the ESR_EN signal is not qualified with a scan enable (SE) signal; as a result, it is impossible to use this solution in a self-test environment without destroying the contents of the scan chains during the shift operation.

[**0091**] **FIG. 3A** shows a block diagram **300** of two set/reset controllers in a design without any ripple structure, in accordance with the present invention. The set/reset controller **303**, controlled by a local scan enable signal SE**1315** and a local set/reset enable signal SR_EN**1316**, consists of a capture controller **305** and a shift controller **306**. The set/reset controller **304**, controlled by a local scan enable signal SE**2317** and a local set/reset enable signal SR_EN**2318**, consists of a capture controller **307** and a shift controller **308**. The local scan enable signals SE**1315** and SE**2317** are driven by a global scan enable signal global_SE **312**. The local set/reset enable signals SR_EN**1316** and SR_EN**2318** are driven by a global set/reset enable signal global_SR_EN **311**. Note that the global scan enable signal global_SE **312** and the global set/reset enable signal global_SR_EN **311** are either generated in the scan-based integrated circuit under test or provided as an input signal to the scan-based integrated circuit. In addition, it is assumed that there is no path from the Q output **326** of the scan cell SC**2310** to the set/reset circuitry **301** and that there is no path from the Q output **325** of the scan cell SC**1309** to the set/reset circuitry **302**. That is, there is no ripple structure existing between the two scan cells SC**1309** and SC**2310**.

[**0092**] A set/reset controller can avoid race conditions and glitches that may arise in the prior-art #3 solution, while preserving its capability of detecting asynchronous set/reset faults in a scan-based integrated circuit. For example, the set/reset controller **303** consists of the capture controller **305** and the shift controller **306**. The set/reset controller **303** provides a new asynchronous set/reset signal **319**, controlled by two enable signals, namely the scan enable SE**1315** and the set/reset enable SR_EN**1316**. The shift controller **306** is used to guarantee that the new asynchronous set/reset signal **319** remains disabled during the shift operation in order to avoid destroying any data that are being shifted into the scan cell **309**. The capture controller **305**, together with the shift controller **306**, is used to realize a two-stage control on the new asynchronous set/reset signal **319** during the capture operation to guarantee that faults present in the original asynchronous set/reset circuitry **301** are detected without any race condition or glitch.

[**0093**] At the first stage of the capture operation, the SR_EN1 signal **316** is set to logic value 0, and capture clocks are applied to capture the test response into all scan cells through their data ports. At this stage, the new asynchronous set/reset signal **319** is disabled, ensuring that no race conditions and glitches arise. At the second stage of the capture operation, the SR_EN1 signal **316** is set to logic value 1 while disabling all capture clocks to allow the faults present in the original asynchronous set/reset circuitry **301** to be propagated via **319** to the scan cell **309**. As a result, the faults present in the original asynchronous set/reset circuitry **301** can be detected.

[**0094**] **FIG. 3B** shows a block diagram **330** of three set/reset controllers in a design with a two-stage ripple structure, in accordance with the present invention.

[**0095**] The set/reset controller **337**, controlled by a local scan enable signal SE**1352** and a local set/reset enable signal SR_EN**1353**, consists of a capture controller **340** and a shift controller **341**. The set/reset controller **338**, controlled by a local scan enable signal SE**2354** and a local set/reset enable signal SR_EN**2355**, consists of a capture controller **342** and a shift controller **343**. The set/reset controller **339**, controlled by the scan enable signal SE**3356** and the set/reset enable signal SR_EN**3357**, consists of a capture controller **344** and a shift controller **345**.

[**0096**] In addition, it is assumed that there is no path from the Q output **368** of the scan cell SC**2335** to the set/reset circuitry **331** and that there is no path from the Q output **367** of the scan cell SC**1334** to the set/reset circuitry **332**. That is, there is no ripple structure existing between the two scan cells SC**1334** and SC**2335**. However, note that the set/reset circuitry **333** accepts inputs from scan cells SC**1334** and SC**2335**. Obviously, this is a two-stage ripple structure. If both SC**1334** and SC**2335** change states simultaneously, possible race conditions may cause glitches to reset the scan cell SC**3336** unexpectedly during test.

[**0097**] To avoid such scenario, two global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are used. The global_SR_EN1 signal **347** is used to drive two local set/reset enable signals SR_EN**1353** and SR_EN**2355** for the scan cells SC**1334** and SC**2335** in the first stage of the ripple structure. The global_SR_EN2 signal **346** is used to drive one local set/reset enable signal SR_EN**3357** for the scan cell SC**3336** in the second stage of the ripple structure. In addition, one global scan enable signal global_SE **348** is used to drive all three local scan enable signals SE**1352**, SE**2354**, and SE**3356**. Note that the global scan enable signal global_SE **348**, the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are either generated in the scan-based integrated circuit under test or provided as an input signal to the scan-based integrated circuit.

[**0098**] During the shift operation, the global_SE signal **348** is set to logic value 1. This will disable the asynchro-

nous set/reset signals **358** to **360** so that the data that are being shifted into the scan cells SC**1334** to SC**3336** will not be destroyed. During the capture operation, clocks CK**1362**, CK**2364**, and CK**3366** are applied first to test data faults propagated via D**1361**, D**2363**, and D**3365**. During data fault testing, global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to disable the asynchronous set/reset signals **358** to **360** for the scan cells SC**1334** to SC**3336** to make sure that the testing of data faults will not be disturbed by the unexpected resetting of any scan cell. After data faults are tested by applying the clocks CK**1362**, CK**2364**, and CK**3366**, the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to allow faults in the set/reset circuitries **331** to **333** to be propagated to the scan cells s SC**1334** to SC**3336**, respectively. Note that the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set in a way that they are not active simultaneously. This is to prevent the state changes of the scan cells SC**1334** and SC**2335** from causing any glitch for the scan cell SC**3336**. As a result, the faults present in the original asynchronous set/reset circuitries **331** to **333** can be detected without any race conditions even in the presence of a ripple structure.

[**0099**] **FIG. 3C** shows an embodiment **370** of a set/reset controller, in accordance with the present invention. The capture controller **376** consists of one inverter **378**. The shift controller **377** consists of one NOR gate **379** and one AND gate **380**. During the shift operation, the scan enable signal SE **382** is set to logic value 1. As a result, the shift controller **375** will set the asynchronous reset signal **392** of the scan cell **381** to logic value 0. That is, the reset capability of the scan cell **381** will be disabled, preventing the data shifted to this scan cell from being destroyed. After the shift operation is completed, the circuit enters the capture operation when the scan enable signal SE **382** is set to logic value 0. At the first stage of the capture operation, the SR_EN signal **383** is set to logic value 0. As a result, the asynchronous reset signal **392** will remain disabled. The capture clock CK **388** is applied to capture the faults present in the functional logic block **372** into the scan cell **381** via its data input port **389**. At the second stage of the capture operation, the capture clock CK **388** is disabled and the SR_EN signal **383** is set to logic value 1. This will set the signal **390** to logic value 1 enabling the propagation of the faults present in the original set/reset circuitry **371** to the scan cell **381** via its asynchronous reset port RESET **392**.

[**0100**] **FIG. 4A** shows a timing diagram **400***a* for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with non-overlapping single-capture clocks. During the first cycle in the capture operation **402***a*, two single pulses are applied to the capture clocks CK**1322** and CK**2324** in a non-overlapping manner as shown at **405***a* and **406***a* to detect data faults while the global set/reset enable global_SR_EN **311** is set to logic value 0. This non-overlapping capture clock scheme is used to avoid the impact of clock skews between two clock domains. Then, during the second cycle in the same capture operation **402***a*, the global set/reset enable global_SR_EN **311** is set to logic value 1 as shown at **404***a* while the capture clocks CK**1322** and CK**2324** are inactive; as a result, set/reset faults are detected.

[**0101**] **FIG. 4B** shows a timing diagram **410***a* for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with overlapping single-capture clocks. During the first cycle in the capture operation **412***a*, two single pulses are applied to the capture clocks CK**1322** and CK**2324** in an overlapping manner as shown at **415***a* and **416***a* to detect data faults while the global set/reset enable global_SR_EN **311** is set to logic value 0. This overlapping capture clock scheme can be used when there is no interaction between two clock domains or clock skews between two clock domains are properly managed. Then, during the second cycle in the same capture operation **412***a*, the global set/reset enable global_SR_EN **311** is set to logic value 1 as shown at **414***a* while the capture clocks CK**1322** and CK**2324** are inactive; as a result, set/reset faults are detected.

[**0102**] **FIG. 4C** shows a timing diagram **420***a* for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with non-overlapping at-speed double-capture clocks. During the first cycle in the capture operation **422***a*, two at-speed double pulses are applied to the capture clocks CK**1322** and CK**2324** in a non-overlapping manner as shown at **425***a* to **428***a* to detect data faults while the global set/reset enable global_SR_EN **311** is set to logic value 0. This non-overlapping capture clock scheme is used to avoid the impact of clock skews between two clock domains. Then, during the second cycle in the same capture operation **422***a*, the global set/reset enable global_SR_EN **311** is set to logic value 1 as shown at **424***a* while the capture clocks CK**1322** and CK**2324** are inactive; as a result, set/reset faults are detected. This timing diagram shows that delay faults in functional logic can be tested with a double-capture approach, in accordance with the present invention. Note that delay faults can also be tested with a single-capture or last-shift-launch approach, in accordance with the present invention.

[**0103**] **FIG. 4D** shows a timing diagram **430***a* for testing the design without any ripple structure shown in **FIG. 3A**, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with overlapping at-speed double-capture clocks. During the first cycle in the capture operation **432***a*, two at-speed double pulses are applied to the capture clocks CK**1322** and CK**2324** in an overlapping manner as shown at **435***a* to **438***a* to detect data faults while the global set/reset enable global_SR_EN **311** is set to logic value 0. This overlapping capture clock scheme can be used when there is no interaction between two clock domains or clock skews between two clock domains are properly managed. Then, during the second cycle in the same capture operation **432***a*, the global set/reset enable global_SR_EN **311** is set to logic value 1 as shown at **434***a* while the capture clocks CK**1322** and CK**2324** are inactive; as a result, set/reset faults are detected. This timing diagram shows that delay faults in functional logic can be tested with a double-capture approach, in accordance with the present invention. Note that delay faults can also be tested with a single-capture or last-shift-launch approach, in accordance with the present invention.

[0104] **FIG. 4E** shows a timing diagram **440***a* for testing the design without any ripple structure shown in **FIG. 3**A, in accordance with the present invention, where data faults and set/reset faults are detected during two capture operations with non-overlapping single-capture clocks. During the first capture operation **442***a* for test pattern i, two single pulses are applied to the capture clocks CK**1322** and CK**2324** as shown at **448***a* and **449***a* while the global set/reset enable global_SR_EN **311** is set to logic value 0 for the whole capture operation, in order for test pattern i to detect data faults. This non-overlapping capture clock scheme is used to avoid the impact of clock skews between two clock domains. Then, during the second capture operation **445***a* for test pattern j, the global set/reset enable global_SR_EN **311** is set to logic value 1 as shown at **447***a* while the capture clocks CK**1322** and CK**2324** are kept inactive for the whole capture operation, in order for test pattern j to detect set/reset faults.

[0105] **FIG. 4F** shows a timing diagram **450***a* for testing the design without any ripple structure shown in **FIG. 3**A, in accordance with the present invention, where data faults and set/reset faults are detected during two capture operations with non-overlapping at-speed double-capture clocks. During the first capture operation **452***a* for test pattern i, two at-speed double pulses are applied to the capture clocks CK**1322** and CK**2324** as shown at **458***a* to **461***a* while the global set/reset enable global_SR_EN **311** is set to logic value 0 for the whole capture operation, in order for test pattern i to detect data faults. This non-overlapping capture clock scheme is used to avoid the impact of clock skews between two clock domains. Then, during the second capture operation **455***a* for test pattern j, the global set/reset enable global_SR_EN **311** is set to logic value 1 as shown at **457***a* while the capture clocks CK**1322** and CK**2324** are kept inactive for the whole capture operation, in order for test pattern j to detect set/reset faults. This timing diagram shows that delay faults in functional logic can be tested with a double-capture approach, in accordance with the present invention. Note that delay faults can also be tested with a single-capture or last-shift-launch approach, in accordance with the present invention.

[0106] **FIG. 4G** shows a timing diagram **400***b* for testing the design with a two-stage ripple structure shown in **FIG. 3**B, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with non-overlapping single-capture clocks. During the first cycle in the capture operation **402***b*, three single pulses are applied to the capture clocks CK**1362**, CK**2364**, and CK**3366** in a non-overlapping manner as shown at **406***b* to **408***b* to detect data faults while the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to logic value 0. This non-overlapping capture clock scheme is used to avoid the impact of clock skews between two clock domains. Then, during the second cycle in the same capture operation, the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to logic value 1 in a non-overlapping manner as shown at **404***b* and **405***b* while the capture clocks CK**1362**, CK**2364**, and CK**3366** are inactive; as a result, set/reset faults are detected. Note that the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are not active at the same time. As a result, any glitch caused by state changes due to the active global set/reset enable signal global_SR_EN**1347** will not affect all scan cells controlled by the global set/reset enable signal global_SR_EN**2346**.

[0107] **FIG. 4H** shows a timing diagram **410***b* for testing the design with a two-stage ripple structure shown in **FIG. 3**B, in accordance with the present invention, where both data faults and set/reset faults are detected during the same capture operation with non-overlapping at-speed double-capture clocks. During the first cycle in the capture operation **412***b*, three at-speed double pulses are applied to the capture clocks CK**1362**, CK**2364**, and CK**3366** in a non-overlapping manner as shown at **416***b* to **421***b* to detect data faults while the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to logic value 0. This non-overlapping capture clock scheme is used to avoid the impact of clock skews between two clock domains. Then, during the second cycle in the same capture operation, the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to logic value 1 in a non-overlapping manner as shown at **414***b* and **415***b* while the capture clocks CK**1362**, CK**2364**, and CK**3366** are inactive; as a result, set/reset faults are detected. Note that the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are not active at the same time. As a result, any glitch caused by state changes due to the active global set/reset enable signal global_SR_EN**1347** will not affect all scan cells controlled by the global set/reset enable signal global_SR_EN**2346**. This timing diagram shows that delay faults in functional logic can be tested with a double-capture approach, in accordance with the present invention. Note that delay faults can also be tested with a single-capture or last-shift-launch approach, in accordance with the present invention.

[0108] **FIG. 4I** shows a timing diagram **430***b* for testing the design with a two-stage ripple structure shown in **FIG. 3**B, in accordance with the present invention, where data faults and set/reset faults are detected during two capture operations with non-overlapping single-capture clocks. During the first capture operation **432***b* for test pattern i, three single pulses are applied to the capture clocks CK**1362**, CK**2364**, and CK**3366** as shown at **439***b* to **441***b* while the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to logic value 0 for the whole capture operation, in order for test pattern i to detect data faults. This non-overlapping capture clock scheme is used to avoid the impact of clock skews between two clock domains. Then, during the second capture operation **435***b* for test pattern j, the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to logic value 1 as shown at **437***b* and **438***b* in a non-overlapping manner while the capture clocks CK**1362**, CK**2364**, and CK**3366** are kept inactive for the whole capture operation, in order for test pattern j to detect set/reset faults. Note that the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are not active at the same time. As a result, any glitch caused by state changes due to the active global set/reset enable signal global_SR_EN**1347** will not affect all scan cells controlled by the global set/reset enable signal global_SR_EN**2346**.

[0109] **FIG. 4J** shows a timing diagram **450***b* for testing the design with a two-stage ripple structure shown in **FIG. 3**B, in accordance with the present invention, where data faults and set/reset faults are detected during two capture

operations with non-overlapping at-speed double-capture clocks. During the first capture operation **452***b* for test pattern i, three at-speed double pulses are applied to the capture clocks CK**1362**, CK**2364**, and CK**3366** as shown at **459***b* to **464***b* while the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to logic value 0 for the whole capture operation, in order for test pattern i to detect data faults. This non-overlapping capture clock scheme is used to avoid the impact of clock skews between two clock domains. Then, during the second capture operation **455***b* for test pattern j, the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are set to logic value 1 as shown at **457***b* and **458***b* while the capture clocks CK**1362**, CK**2364**, and CK**3366** are kept inactive for the whole capture operation, in order for test pattern j to detect set/reset faults. Note that the global set/reset enable signals global_SR_EN**1347** and global_SR_EN**2346** are not active at the same time. As a result, any glitch caused by state changes due to the active global set/reset enable signal global_SR_EN**1347** will not affect all scan cells controlled by the global set/reset enable signal global_SR_EN**2346**. This timing diagram shows that delay faults in functional logic can be tested with a double-capture approach, in accordance with the present invention. Note that delay faults can also be tested with a single-capture or last-shift-launch approach, in accordance with the present invention.

[**0110**] **FIG. 5A** shows an example set **500** of RTL (register-transfer level) Verilog codes before and after a sequentially-gated reset violation and a combinationally-gated reset violation are repaired, in accordance with the present invention.

[**0111**] In the original RTL Verilog code, the asynchronous reset signal s_rst on line **11**, of the D flip-flop inferred for signal **q1** in the always block starting from line **11**, can be traced back to the output of the D flip-flop inferred for signal z in the always block starting from line **7**. Note that z is gated with the specified reset source signal rst on line **5** and the result is the asynchronous reset signal s_rst on line **5**. As a result, this is a sequentially-gated reset violation. On the other hand, the asynchronous reset signal c_rst on line **18**, of the D flip-flop inferred for signal **q2** in the always block starting from line **18**, can be traced back to the primary input x on line **6**. Note that x is gated with the specified reset source rst on line **6** and the result is the asynchronous reset signal c_rst on line **6**. As a result, this is a combinationally-gated reset violation.

[**0112**] In the modified RTL Verilog code, two new signals, scan_s_rst on line **6** and scan_c_rst on line **7**, are added to model the repaired s_rst and c_rst signals, respectively. The continuous assignment statements on lines **10** and **12** describe the set/reset controllers that are inserted to repair the sequentially-gated reset violation and the combinationally-gated reset violation, respectively. When SE is set to logic value 0 and SR_EN is set to logic value 1, the modified circuit behavior is the same as the original one. When SE has logic value 1, scan_s_rst and scan_c_rst will become logic value 0, thus disabling the asynchronous reset operation of the D flip-flops inferred for signals q1 and q2 in the always blocks starting from lines **20** and **27**, respectively.

[**0113**] **FIG. 5B** shows an example set of RTL (register-transfer level) Verilog codes **510** before and after a generated reset violation and a destructive reset violation are repaired, in accordance with the present invention.

[**0114**] In the original RTL Verilog code, the asynchronous reset signal g_rst on line **10**, of the D flip-flop inferred for signal **q1** in the always block starting from line **10**, can be traced back to the output of the D-flip flop inferred for g_rst described in the always block starting from line **6**. As a result, this is a generated reset violation. On the other hand, the asynchronous reset signal d_rst on line **17**, of the D flip-flop inferred for signal **q2** in the always block starting from line **17**, is always stuck at logic value 1. As a result, this is a destructive reset violation because the D flip-flop inferred for signal **q2** in the always block starting from line **17** will always be reset.

[**0115**] In the modified RTL Verilog code, two new signals, scan_g_rst on line **6** and scan_d_rst on line **7**, are added to model the repaired g_rst and d_rst signals, respectively. The continuous assignment statements on lines **10** and **12** model the added set/reset controllers that repair the generated reset violation and the destructive reset violation, respectively. When SE has logic value 0 and SR_EN is set logic value 1, the RTL circuit behavior is the same as the original one; when SE has logic value 1, the signal scan_g_rst and scan_d_rst will become logic value 0, thus disabling the asynchronous reset operation of the D flip-flops inferred for signals q1 and q2 in the always block starting from lines **19** and **26**, respectively.

[**0116**] FIG. SC shows the gate-level circuit model **520** corresponding to the original RTL (register-transfer level) code shown in **FIG. 5A**. D flip-flops DFF**2522** and DFF**3523** are reset by asynchronous signals s_rst **531** and c_rst **532**, respectively. Since the value of s_rst **531** is determined by an AND gate **524** with the output z **530** of the D flip-flop DFF**1521** as one of its inputs, this is a sequentially-gated reset violation. Since the value of c_rst **532** is determined by an AND gate **525** with only primary inputs rst **526** and x **527** as its inputs, this is a combinationally-gated reset violation.

[**0117**] **FIG. 5D** shows the gate-level circuit model **540** obtained after the sequentially-gated reset violation and the combinationally-gated reset violation shown in **FIG. 5C** are repaired, in accordance with the present invention. The set/reset controllers that are added to disable the reset operations of D flip-flops DFF**2522** and DFF**3523** consist of two AND gates **541** and **542**, one inverter **543**, and one NOR gate **544**.

[**0118**] In functional mode, SE **545** has logic value 0 and SR_EN **546** has logic value 1. As a result, the original reset signals s_rst **531** and c_rst **532** will be able to reset DFF**2522** and DFF**3523**, respectively, as intended by the functionality of the circuit.

[**0119**] During the shift operation, SE **545** is set to logic value 1 while SR_EN **546** may take any logic value. As a result, the new reset signals scan_s_rst **548** and scan_c_rst **549** will become logic value 0, preventing DFF**2522** and DFF**3523** from being reset during the shift operation, respectively. Therefore, the shift operation can be conducted correctly.

[**0120**] During the capture operation, SE **545** is set to logic value 0. In the first stage of the capture operation, SR_EN **546** is set to logic value 0 and the capture clock ck **529** is

applied to capture the faults from the signal line d **528** into DFF**2522** and DFF**3523**. In the second stage of the capture operation, the capture clock ck **529** is disabled and SR_EN **546** is set to logic value 1. As a result, the logic value of signal **547** becomes logic value 1, which allows the faults from the original reset signals s_rst **531** and c_rst **532** to be propagated to DFF**2522** and DFF**3523**, respectively. Therefore, fault coverage can be improved without any race condition or glitch.

[**0121**] FIG. SE shows the gate-level circuit model **560** corresponding to the original RTL (register-transfer level) code shown in **FIG. 5B**. D flip-flops DFF**2562** and DFF**3563** are reset by asynchronous signals g_rst **567** and d_rst **568**, respectively. Since the reset signal g_jst **567** of DFF**2562** comes directly from the D flip-flop DFF**1561**, this is a generated reset violation. Since the reset signal d_rst **568** of DFF**3563** is tied to VCC (logic value 1), this is a destructive reset violation.

[**0122**] **FIG. 5F** shows the gate-level circuit model **580** after the generated reset violation and the destructive reset violation shown in FIG. SE are repaired, in accordance with the present invention. The set/reset controllers that are added to disable the reset operations of D flip-flops DFF**2562** and DFF**3563** consist of two AND gates **581** and **582**, one inverter **583**, and one NOR gate **584**.

[**0123**] In functional mode, SE **585** has logic value 0 and SR_EN **586** has logic value 1. As a result, the original reset signals g_rst **567** and d_rst **568** will be able to reset DFF**2562** and DFF**3563**, respectively, as intended by the functionality of the circuit.

[**0124**] During the shift operation, SE **585** is set to logic value 1 while SR_EN **586** may take any logic value. As a result, the new reset signals scan_g_rst **588** and scan_d_rst **589** will become logic value 0, preventing DFF**2562** and DFF**3563** from being reset during the shift operation, respectively. Therefore, the shift operation can be conducted correctly.

[**0125**] During the capture operation, SE **585** is set to logic value 0. In the first stage of the capture operation, SR_EN **586** is set to logic value 0 and the capture clock ck **566** is applied to capture the faults from the signal line d **565** into DFF**2562** and DFF**3563**. In the second stage of the capture operation, the capture clock ck **566** is disabled and SR_EN **586** is set to logic value 1. The logic value of the signal **587** becomes logic value 1, allowing the faults from the original reset signals g_rst **567** and d_rst **568** to be propagated to DFF**2562** and DFF**3563**, respectively. Therefore, fault coverage can be improved without any race condition or glitch.

[**0126**] **FIG. 6** shows a flow diagram **600** of the method for repairing asynchronous set/reset violations at either RTL (register-transfer level) or gate-level, in accordance with the present invention. The system **600**, which consists of a number of computer-implemented steps, accepts the user-supplied synthesizable RTL or gate-level HDL (hardware design language) code **601** representing a scan-based integrated circuit design, the control files **602**, a chosen foundry library **603**, and an asynchronous set/reset signal list **604**. The control files **602** contain all set-up information and scripts to control the steps of compiling **605** the HDL code **601** into a sequential circuit model **606** and automatic set/reset controller synthesis **607** at either RTL or gate-level. The automatic set/reset controller synthesis **607** produces repaired RTL or gate-level HDL code **608**, which contains set/reset controllers added to repair all asynchronous set/reset signals specified by the list **604**. All reports and errors are stored in the report files **609**.

[**0127**] **FIG. 7A** shows a flow diagram **700** of the method for generating test patterns for data faults and set/reset faults in scan-test mode, in accordance with the present invention. The system **700** accepts the user-supplied RTL (register-transfer level) or gate-level HDL (hardware design language) code **701** representing a scan-based integrated circuit design whose asynchronous set/reset violations have been repaired. In addition, control files **702**, a chosen foundry library **703**, and an input constraint file **704** are also provided. The input constraint file **704** contains input constraints on all clocks, set/reset enable (SR_EN) signals, and scan enable (SE) signals. The control files **702** contain all set-up information and scripts required for compilation **705**, model transformation **707**, predetermined pattern fault simulation **709**, combinational ATPG (automatic test pattern generation) **710**, and post-processing **711**. The compilation step **705** is to compile the HDL code **701** into a sequential circuit model **706**. The model transformation step **707** is to convert the sequential circuit model **706** into an equivalent combinational circuit model **708**. The predetermined pattern fault simulation step **709** is to identify the faults that are detected by a set of predetermined patterns. The combinational ATPG (automatic test pattern generation) step **710** is to generate test patterns for testing data faults and set/reset faults. Finally, the post-processing step **711** is to generate HDL test benches and ATE (automatic test equipment) test programs **712**. All reports and errors are stored in the report files **713**.

[**0128**] **FIG. 7B** shows a flow diagram **750** of the method for generating test patterns for data faults and set/reset faults in self-test mode, in accordance with the present invention. The system **750** accepts the user-supplied RTL (register-transfer level) or gate-level HDL (hardware design language) code **751** representing a scan-based integrated circuit design whose asynchronous set/reset violations have been repaired. In addition, control files **752**, a chosen foundry library **753**, and an input constraint file **754** are also provided. The input constraint file **754** contains input constraints on all clocks, set/reset enable (SR_EN) signals, and scan enable (SE) signals. The control files **752** contain all set-up information and scripts required for compilation **755**, model transformation **757**, pseudo-random pattern fault simulation **759**, and post-processing **760**. The compilation step **755** is to compile the HDL code **701** into a sequential circuit model **756**. The model transformation step **757** is to convert the sequential circuit model **756** into an equivalent combinational circuit model **758**. The pseudo-random pattern fault simulation step **759** is to identify the faults that are detected by a set of pseudo-random patterns. Finally, the post-processing step **760** is to generate HDL test benches and ATE (automatic test equipment) test programs **761**. All reports and errors are stored in the report files **762**.

[**0129**] **FIG. 8** shows an example electronic design automation system **800** in which the method for repairing asynchronous set/reset violations at either RTL (register-transfer level) or gate-level and the method of generating test patterns for data faults and set/reset faults, in accordance with the present invention, may be implemented. The system

800 includes a processor **802**, which operates together with a memory **801** to run a set of the asynchronous set/reset repair and test pattern generation software. The processor **802** may represent a central processing unit of a personal computer, workstation, mainframe computer or other suitable digital processing device. The memory **801** can be an electronic memory or a magnetic or optical disk-based memory, or various combinations thereof. A designer interacts with the asynchronous set/reset repair and test pattern generation software run by the processor **802** to provide appropriate inputs via an input device **803**, which may be a keyboard, disk drive or other suitable source of design information. The processor **802** provides outputs to the designer via an output device **804**, which may be a display, a printer, a disk drive or various combinations of these and other elements.

[0130] Having thus described presently preferred embodiments of the present invention, it can now be appreciated that the objectives of the invention have been fully achieved. And it will be understood by those skilled in the art that many changes in construction & circuitry, and widely differing embodiments & applications of the invention will suggest themselves without departing from the spirit and scope of the present invention. The disclosures and the description herein are intended to be illustrative and are not in any sense limitation of the invention, more preferably defined in scope by the following claims.

What is claimed is:

1. A method for testing faults propagated to the data ports and asynchronous set/reset ports of selected scan cells in a scan-based integrated circuit in a selected scan-test mode or selected self-test mode, the scan-based integrated circuit containing a plurality of set/reset circuitries, a plurality of set/reset controllers, and a plurality of scan chains, each scan chain comprising multiple scan cells coupled in series, each scan cell having one or more clocks and each set/reset controller having a scan enable (SE) signal and a set/reset enable (SR_EN) signal; said method comprising:

(a) shifting in a stimulus to all said scan cells in said scan-based integrated circuit by enabling all said scan enable (SE) signals connected to all said scan cells during a shift-in operation;

(b) capturing a test response of all said scan cells for testing said faults propagated to said data ports and said asynchronous set/reset ports of all said selected scan cells by enabling or disabling all said set/reset enable (SR_EN) signals connected to all said selected scan cells during a capture operation;

(c) shifting out said test response for comparison or compaction while shifting in a new stimulus to all said scan cells during a shift-out operation; and

(d) repeating steps (b) to (c) until a limiting criteria is reached.

2. The method of claim 1, wherein said shifting in a stimulus to all said scan cells further comprises selectively shifting in a predetermined stimulus from an ATE (automatic test equipment) in said selected scan-test mode or shifting in a pseudo-random stimulus automatically generated in said scan-based integrated circuit using a pseudo-random pattern generator (PRPG) in said selected self-test mode during said shift-in operation.

3. The method of claim 1, wherein said shifting in a stimulus to all said scan cells further comprises using all said set/reset enable (SR_EN) signals to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in operation.

4. The method of claim 1, wherein said capturing a test response of all said scan cells further comprises selectively disabling all said scan enable (SE) signals simultaneously or in an ordered sequence during said capture operation.

5. The method of claim 1, wherein said capturing a test response of all said scan cells further comprises disabling all said clocks controlling all said scan cells, while enabling all said set/reset enable (SR_EN) signals, for testing said faults propagated to said asynchronous set/reset ports of all said selected scan cells during said capture operation.

6. The method of claim 5, wherein said enabling all said set/reset enable (SR_EN) signals further comprises selectively enabling two or more said set/reset enable (SR_EN) signals simultaneously or in an ordered sequence during said capture operation.

7. The method of claim 1, wherein said capturing a test response of all said scan cells further comprises enabling all said clocks controlling all said scan cells, while disabling all said set/reset enable (SR_EN) signals, for testing said faults propagated to said data ports of all said selected scan cells during said capture operation.

8. The method of claim 7, wherein said enabling all said clocks controlling all said scan cells further comprises selectively enabling two or more said clocks controlling two or more said scan cells simultaneously or in an ordered sequence during said capture operation.

9. The method of claim 1, wherein said shifting out said test response for comparison or compaction further comprises selectively shifting out said test response to said ATE for comparison in said selected scan-test mode or shifting out said test response for compaction using a compactor, including a multiple-input signature register (MISR), in said selected self-test mode during said shift-out operation.

10. The method of claim 1, wherein said set/reset controller further comprises providing a shift controller and a capture controller in response to a said scan enable (SE) signal and a said set/reset enable (SR_EN) signal, wherein said shift controller is adapted to disable said asynchronous set/reset ports of one or more said selected scan cells during said shift-in or said shift-out operation, and wherein said capture controller is adapted to enable or disable propagation of said faults present in one said set/reset circuitry to said asynchronous set/reset ports of one or more said selected scan cells during said capture operation.

11. The method of claim 1, wherein said scan enable (SE) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

12. The method of claim 11, wherein all said scan enable (SE) signals are further driven by one or more global scan enable (global_SE) signals, wherein each said global scan enable (global_SE) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

13. The method of claim 1, wherein said set/reset enable (SR_EN) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

14. The method of claim 13, wherein all set/reset enable (SR_EN) signals are further driven by one or more global set/reset enable (global_SR_EN) signals, wherein each said global set/reset enable (global_SR_EN) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

15. The method of claim 1, wherein said scan cell is a multiplexed-type D flip-flop, a two-port D flip-flop, or a LSSD (level-sensitive scan design) SRL (shift register latch).

16. The method of claim 1, wherein said set/reset controller is used to repair one or more asynchronous set/reset violations, comprising sequentially-gated set/reset violations, combinationally-gated set/reset violations, generated set/reset violations, and destructive set/reset violations, in a selected set/reset circuitry in said scan-based integrated circuit.

17. A set/reset controller having a scan enable (SE) signal and a set/reset enable (SR_EN) signal for testing faults propagated to the data ports and asynchronous set/reset ports of selected scan cells in a scan-based integrated circuit, the scan-based integrated circuit containing a plurality of set/reset circuitries and a plurality of scan chains, each scan chain comprising multiple scan cells coupled in series, each scan cell having one or more clocks; said set/reset controller comprising:

(a) a shift controller, inserted between a selected set/reset circuitry and said asynchronous set/reset ports of all said selected scan cells, for disabling said asynchronous set/reset ports of all said selected scan cells, in response to said scan enable (SE) signal and said set/reset enable (SR_EN) signal, during a shift-in or shift-out operation; and

(b) a capture controller, inserted between said selected set/reset circuitry and said asynchronous set/reset ports of all said selected scan cells, for enabling or disabling propagation of said faults present in said selected set/reset circuitry to said asynchronous set/reset ports of all said selected scan cells, in response to said scan enable (SE) signal and said set/reset enable (SR_EN) signal, during a capture operation.

18. The set/reset controller of claim 17, wherein said shift controller is selectively embedded in said selected set/reset circuitry or in all said selected scan cells, and wherein said enable (SE) signal, said set/reset enable (SR_EN) signal, or said scan enable (SE) signal and said set/reset enable (SR_EN) signal can be selectively used to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in or shift-out operation.

19. The set/reset controller of claim 17, wherein said capture controller further comprises selectively disabling all said scan enable (SE) signals simultaneously or in an ordered sequence during said capture operation.

20. The set/reset controller of claim 17, wherein said capture controller further comprises disabling all said clocks controlling all said scan cells, while enabling all said set/reset enable (SR_EN) signals, for testing said faults propagated to said asynchronous set/reset ports of all said selected scan cells during said capture operation.

21. The set/reset controller of claim 20, wherein said enabling all said set/reset enable (SR_EN) signals further comprises selectively enabling two or more said set/reset enable (SR_EN) signals simultaneously or in an ordered sequence during said capture operation.

22. The set/reset controller of claim 17, wherein said capture controller further comprises enabling all said clocks controlling all said scan cells, while disabling all said set/reset enable (SR_EN) signals, for testing said faults propagated to said data ports of all said selected scan cells during said capture operation.

23. The set/reset controller of claim 22, wherein said enabling all said clocks controlling all said scan cells further comprises selectively enabling two or more said clocks controlling two or more said scan cells simultaneously or in an ordered sequence during said capture operation.

24. The set/reset controller of claim 17, wherein said capture controller is selectively embedded in said selected set/reset circuitry or in all said selected scan cells.

25. The set/reset controller of claim 17, wherein said scan enable (SE) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

26. The set/reset controller of claim 25, wherein all said scan enable (SE) signals are further driven by one or more global scan enable (global_SE) signals, wherein each said global scan enable (global_SE) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

27. The set/reset controller of claim 17, wherein said set/reset enable (SR_EN) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

28. The set/reset controller of claim 27, wherein all set/reset enable (SR_EN) signals are further driven by one or more global set/reset enable (global_SR_EN) signals, wherein each said global set/reset enable (global_SR_EN) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

29. The set/reset controller of claim 17, wherein said scan cell is a multiplexed-type D flip-flop, a two-port D flip-flop, or a LSSD (level-sensitive scan design) SRL (shift register latch).

30. The set/reset controller of claim 17, wherein said shift controller and capture controller are used to repair one or more asynchronous set/reset violations, comprising sequentially-gated set/reset violations, combinationally-gated set/reset violations, generated set/reset violations, and destructive set/reset violations, in said selected set/reset circuitry in said scan-based integrated circuit.

31. A method for synthesizing a plurality of set/reset controllers each having a scan enable (SE) signal and a set/reset enable (SR_EN) signal for testing faults propagated to the data ports and asynchronous set/reset ports of selected scan cells in a scan-based integrated circuit, the scan-based integrated circuit containing a plurality of set/reset circuitries and a plurality of scan chains, each scan chain comprising multiple scan cells coupled in series, each scan cell having one or more clocks; said method comprising the computer-implemented steps of:

(a) compiling the HDL (hardware description language) code modeled at RTL (register-transfer level) or gate-level that represents said scan-based integrated circuit into a sequential circuit model;

(b) specifying a list of asynchronous set/reset signals each causing one or more asynchronous set/reset violations in each selected set/reset circuitry for repair;

(c) synthesizing a said plurality of set/reset controllers, each having a said scan enable (SE) signal and a said set/reset enable (SR_EN) signal, on said sequential circuit model according to said list of asynchronous set/reset signals; and

(d) generating the repaired HDL code in a selected RTL or gate-level format.

32. The method of claim 32, wherein said specifying a list of asynchronous set/reset signals further comprises automatically identifying said list of asynchronous set/reset signals using simulation methods.

33. The method of claim 32, wherein said set/reset controller in said synthesizing a said plurality of set/reset controllers further comprises

(e) a shift controller, inserted between said selected set/reset circuitry and said asynchronous set/reset ports of all said selected scan cells, for disabling said asynchronous set/reset ports of all said selected scan cells, in response to said scan enable (SE) signal and said set/reset enable (SR_EN) signal, during a shift-in or shift-out operation; and

(f) a capture controller, inserted between said selected set/reset circuitry and said asynchronous set/reset ports of all said selected scan cells, for enabling or disabling propagation of said faults present in said selected set/reset circuitry to said asynchronous set/reset ports of one or more said selected scan cells, in response to said scan enable (SE) signal and said set/reset enable (SR_EN) signal, during a capture operation.

34. The method of claim 33, wherein said shift controller is selectively embedded in said selected set/reset circuitry or in one or more said selected scan cells, and wherein said enable (SE) signal, said set/reset enable (SR_EN) signal, or said scan enable (SE) signal and said set/reset enable (SR_EN) signal can be selectively used to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in or said shift-out operation.

35. The method of claim 33, wherein said capture controller further comprises selectively disabling all said scan enable (SE) signals simultaneously or in an ordered sequence during said capture operation.

36. The method of claim 33, wherein said capture controller further comprises disabling all said clocks controlling all said scan cells, while enabling all said set/reset enable (SR_EN) signals, for testing said faults propagated to said asynchronous set/reset ports of one or more said selected scan cells during said capture operation.

37. The method of claim 36, wherein said enabling all said set/reset enable (SR_EN) signals further comprises selectively enabling two or more said set/reset enable (SR_EN) signals simultaneously or in an ordered sequence during said capture operation.

38. The method of claim 33, wherein said capture controller further comprises enabling all said clocks controlling all said scan cells, while disabling all said set/reset enable (SR_EN) signals, for testing said faults propagated to said data ports of all said selected scan cells during said capture operation.

39. The method of claim 38, wherein said enabling all said clocks controlling all said scan cells further comprises selectively enabling two or more said clocks controlling two or more said selected scan cells simultaneously or in an ordered sequence during said capture operation.

40. The method of claim 33, wherein said capture controller is selectively embedded in said selected set/reset circuitry or in said selected scan cells.

41. The method of claim 33, wherein said scan enable (SE) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

42. The method of claim 41, wherein all said scan enable (SE) signals are further driven by one or more global scan enable (global_SE) signals, wherein each said global scan enable (global_SE) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

43. The method of claim 33, wherein said set/reset enable (SR_EN) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

44. The method of claim 43, wherein all set/reset enable (SR_EN) signals are further driven by one or more global set/reset enable (global_SR_EN) signals, wherein each said global set/reset enable (global_SR_EN) signal is selectively generated in said scan-based integrated circuit or an input signal to said scan-based integrated circuit.

45. The method of claim 33, wherein said scan cell is a multiplexed-type D flip-flop, a two-port D flip-flop, or a LSSD (level-sensitive scan design) SRL (shift register latch).

46. The method of claim 33, wherein said shift controller and capture controller are used to repair one or more asynchronous set/reset violations, comprising sequentially-gated set/reset violations, combinationally-gated set/reset violations, generated set/reset violations, and destructive set/reset violations, in said selected set/reset circuitry in said scan-based integrated circuit.

47. A method for generating stimuli and test responses for testing faults propagated to the data ports and asynchronous set/reset ports of selected scan cells in a scan-based integrated circuit in a selected scan-test mode or selected self-test mode, the scan-based integrated circuit containing a plurality of set/reset circuitries, a plurality of set/reset controllers, and a plurality of scan chains, each scan chain comprising multiple scan cells coupled in series, each scan cell having one or more clocks and each set/reset controller having a scan enable (SE) signal and a set/reset enable (SR_EN) signal; said method comprising the computer-implemented steps of:

(a) compiling the HDL (hardware description language) code modeled at RTL (register-transfer level) or gate-level that represents said scan-based integrated circuit into a sequential circuit model;

(b) specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals during a shift-in, capture, or shift-out operation;

(c) transforming said sequential circuit model into an equivalent combinational circuit model; and

(d) generating said stimuli and said test responses according to said input constraints and said combinational circuit model.

48. The method of claim 47, wherein said specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said set/reset enable (SR_EN) signals to logic value 1 to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in or said shift-out operation.

49. The method of claim 47, wherein said specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said scan enable (SE) signals to logic value 1 to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in or said shift-out operation.

50. The method of claim 47, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises selectively setting all said scan enable (SE) signals to logic value 0 simultaneously or in an ordered sequence during said capture operation.

51. The method of claim 47, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said clocks controlling all said scan cells to logic value 0, while setting all said set/reset enable (SR_EN) signals to logic value 1, for testing said faults propagated to said asynchronous set/reset ports of all said selected scan cells during said capture operation.

52. The method of claim 51, wherein said setting all said set/reset enable (SR_EN) signals to logic value 1 further comprises selectively setting two or more said set/reset enable (SR_EN) signals to logic value 1 simultaneously or in an ordered sequence during said capture operation.

53. The method of claim 47, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said clocks controlling all said scan cells to logic value 1, while setting all said set/reset enable (SR_EN) signals to logic value 0, for testing said faults propagated to said data ports of all said selected scan cells during said capture operation.

54. The method of claim 53, wherein said setting all said clocks controlling all said scan cells to logic value 1 further comprises selectively setting two or more said clocks controlling two or more said scan cells to logic value 1 simultaneously or in an ordered sequence during said capture operation.

55. The method of claim 47, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises performing fault simulation on said combinational circuit model using a selected set of predetermined patterns in said selected scan-test mode or a selected set of pseudo-random patterns in said selected self-test mode.

56. The method of claim 47, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises performing combinational ATPG (automatic test pattern generation) on said combinational circuit model to generate said stimuli and said test responses in said selected scan-test mode.

57. The method of claim 47, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises generating HDL test benches according to said stimuli and said test responses for verifying the correctness of said scan-based integrated circuit using simulation methods.

58. The method of claim 47, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises generating ATE (automatic test equipment) test programs according to said stimuli and said test responses for verifying the correctness of said scan-based integrated circuit in said ATE.

59. A computer-readable memory having computer-readable program code embodied therein for causing a computer system to perform a method for generating stimuli and test responses for testing faults propagated to the data ports and asynchronous set/reset ports of selected scan cells in a scan-based integrated circuit in a selected scan-test mode or selected self-test mode, the scan-based integrated circuit containing a plurality of set/reset circuitries, a plurality of set/reset controllers, and a plurality of scan chains, each scan chain comprising multiple scan cells coupled in series, each scan cell having one or more clocks and each set/reset controller having a scan enable (SE) signal and a set/reset enable (SR_EN) signal; said method comprising the computer-implemented steps of:

(a) compiling the HDL (hardware description language) code modeled at RTL (register-transfer level) or gate-level that represents said scan-based integrated circuit into a sequential circuit model;

(b) specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals during a shift-in, capture, or shift-out operation;

(c) transforming said sequential circuit model into an equivalent combinational circuit model; and

(d) generating said stimuli and said test responses according to said input constraints and said combinational circuit model.

60. The computer-readable memory of claim 59, wherein said specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said set/reset enable (SR_EN) signals to logic value 1 to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in or said shift-out operation.

61. The computer-readable memory of claim 59, wherein said specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said scan enable (SE) signals to logic value 1 to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in or said shift-out operation.

62. The computer-readable memory of claim 59, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises selectively setting all said scan enable (SE) signals to logic value 0 simultaneously or in an ordered sequence during said capture operation.

63. The computer-readable memory of claim 59, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said clocks controlling all said scan cells to logic value 0, while setting all said set/reset

enable (SR_EN) signals to logic value 1, for testing said faults propagated to said asynchronous set/reset ports of all said selected scan cells during said capture operation.

64. The computer-readable memory of claim 63, wherein said setting all said set/reset enable (SR_EN) signals to logic value 1 further comprises selectively setting two or more said set/reset enable (SR_EN) signals to logic value 1 simultaneously or in an ordered sequence during said capture operation.

65. The computer-readable memory of claim 59, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said clocks controlling all said scan cells to logic value 1, while setting all said set/reset enable (SR_EN) signals to logic value 0, for testing said faults propagated to said data ports of all said selected scan cells during said capture operation.

66. The computer-readable memory of claim 65, wherein said setting all said clocks controlling all said scan cells to logic value 1 further comprises selectively setting two or more said clocks controlling two or more said scan cells to logic value 1 simultaneously or in an ordered sequence during said capture operation.

67. The computer-readable memory of claim 59, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises performing fault simulation on said combinational circuit model using a selected set of predetermined patterns in said selected scan-test mode or a selected set of pseudo-random patterns in said selected self-test mode.

68. The computer-readable memory of claim 59, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises performing combinational ATPG (automatic test pattern generation) on said combinational circuit model to generate said stimuli and said test responses in said selected scan-test mode.

69. The computer-readable memory of claim 59, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises generating HDL test benches according to said stimuli and said test responses for verifying the correctness of said scan-based integrated circuit using simulation methods.

70. The computer-readable memory of claim 59, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises generating ATE (automatic test equipment) test programs according to said stimuli and said test responses for verifying the correctness of said scan-based integrated circuit in said ATE.

71. An electronic design automation system comprising: a processor; a bus coupled to said processor; and a computer-readable memory coupled to said bus and having computer-readable program code stored therein for causing said electronic design automation system to perform a method for generating stimuli and test responses for testing faults propagated to the data ports and asynchronous set/reset ports of selected scan cells in a scan-based integrated circuit in a selected scan-test mode or selected self-test mode, the scan-based integrated circuit containing a plurality of set/reset circuitries, a plurality of set/reset controllers, and a plurality of scan chains, each scan chain comprising multiple scan cells coupled in series, each scan cell having one or more clocks and each set/reset controller having a scan enable (SE) signal and a set/reset enable (SR_EN) signal; said method comprising the computer-implemented steps of:

(a) compiling the HDL (hardware description language) code modeled at RTL (register-transfer level) or gate-level that represents said scan-based integrated circuit into a sequential circuit model;

(b) specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals during a shift-in, capture, or shift-out operation;

(c) transforming said sequential circuit model into an equivalent combinational circuit model; and

(d) generating said stimuli and said test responses according to said input constraints and said combinational circuit model.

72. The system of claim 71, wherein said specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said set/reset enable (SR_EN) signals to logic value 1 to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in or said shift-out operation.

73. The system of claim 71, wherein said specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said scan enable (SE) signals to logic value 1 to disable all said asynchronous set/reset ports of all said selected scan cells during said shift-in or said shift-out operation.

74. The system of claim 71, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises selectively setting all said scan enable (SE) signals to logic value 0 simultaneously or in an ordered sequence during said capture operation.

75. The system of claim 71, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said clocks controlling all said scan cells to logic value 0, while setting all said set/reset enable (SR_EN) signals to logic value 1, for testing said faults propagated to said asynchronous set/reset ports of all said selected scan cells during said capture operation.

76. The system of claim 75, wherein said setting all said set/reset enable (SR_EN) signals to logic value 1 further comprises selectively setting two or more said set/reset enable (SR_EN) signals to logic value 1 simultaneously or in an ordered sequence during said capture operation.

77. The system of claim 71, wherein specifying input constraints on said clocks, said scan enable (SE) signals, and said set/reset enable (SR_EN) signals further comprises setting all said clocks controlling all said scan cells to logic value 1, while setting all said set/reset enable (SR_EN) signals to logic value 0, for testing said faults propagated to said data ports of all said selected scan cells during said capture operation.

78. The system of claim 77, wherein said setting all said clocks controlling all said scan cells to logic value 1 further comprises selectively setting two or more said clocks con-

trolling two or more said scan cells to logic value 1 simultaneously or in an ordered sequence during said capture operation.

79. The system of claim 71, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises performing fault simulation on said combinational circuit model using a selected set of predetermined patterns in said selected scan-test mode or a selected set of pseudo-random patterns in said selected self-test mode.

80. The system of claim 71, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises performing combinational ATPG (automatic test pattern generation) on said combinational circuit model to generate said stimuli and said test responses in said selected scan-test mode.

81. The system of claim 71, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises generating HDL test benches according to said stimuli and said test responses for verifying the correctness of said scan-based integrated circuit using simulation methods.

82. The system of claim 71, wherein said generating said stimuli and said test responses according to said input constraints and said combinational circuit model further comprises generating ATE (automatic test equipment) test programs according to said stimuli and said test responses for verifying the correctness of said scan-based integrated circuit in said ATE.

* * * * *

US 20020120896A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2002/0120896 A1

Wang et al. (43) Pub. Date: Aug. 29, 2002

(54) MULTIPLE-CAPTURE DFT SYSTEM FOR DETECTING OR LOCATING CROSSING CLOCK-DOMAIN FAULTS DURING SELF-TEST OR SCAN-TEST

(76) Inventors: **Laung-Terng Wang**, Sunnyvale, CA (US); **Po-Ching Hsu**, Hsinchu (TW); **Shih-Chia Kao**, Taipei (TW); **Meng-Chyi Lin**, Taoyuan (TW); **Hsin-Po Wang**, Hsinchu (TW); **Hao-Jan Chao**, Taoyuan (TW); **Xiaoqing Wen**, Sunnyvale, CA (US)

Correspondence Address:
Law Office of Jim Zegeer
Suite 108
801 North Pitt Street
Alexandria, VA 22314 (US)

(21) Appl. No.: **10/067,372**

(22) Filed: **Feb. 7, 2002**

Related U.S. Application Data

(60) Provisional application No. 60/268,601, filed on Feb. 15, 2001.

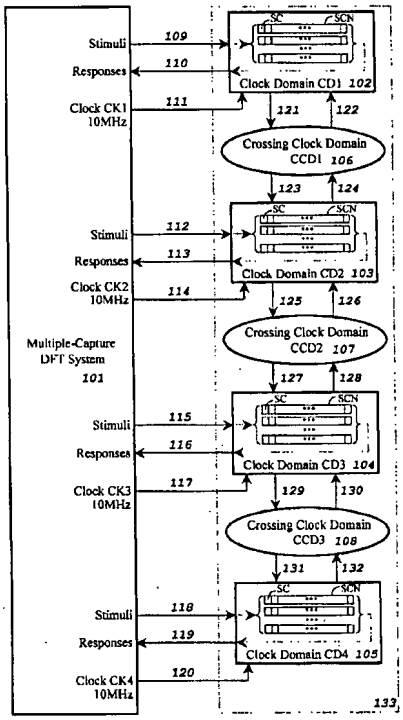
Publication Classification

(51) **Int. Cl.**[7] **G01R 31/28**

(52) **U.S. Cl.** **714/731**

(57) **ABSTRACT**

A method and apparatus for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in self-test or scan-test mode, where N>1 and each domain has a plurality of scan cells. The method and apparatus allows generating and loading N pseudorandom or predetermined stimuli to all the scan cells within the N clock domains in the integrated circuit or circuit assembly during the shift operation, applying an ordered sequence of capture clocks to all the scan cells within the N clock domains during the capture operation, compacting or comparing N output responses of all the scan cells for analysis during the compact/compare operation, and repeating the above process until a predetermined limiting criteria is reached. A computer-aided design (CAD) system is further developed to realize the method and synthesize the apparatus.



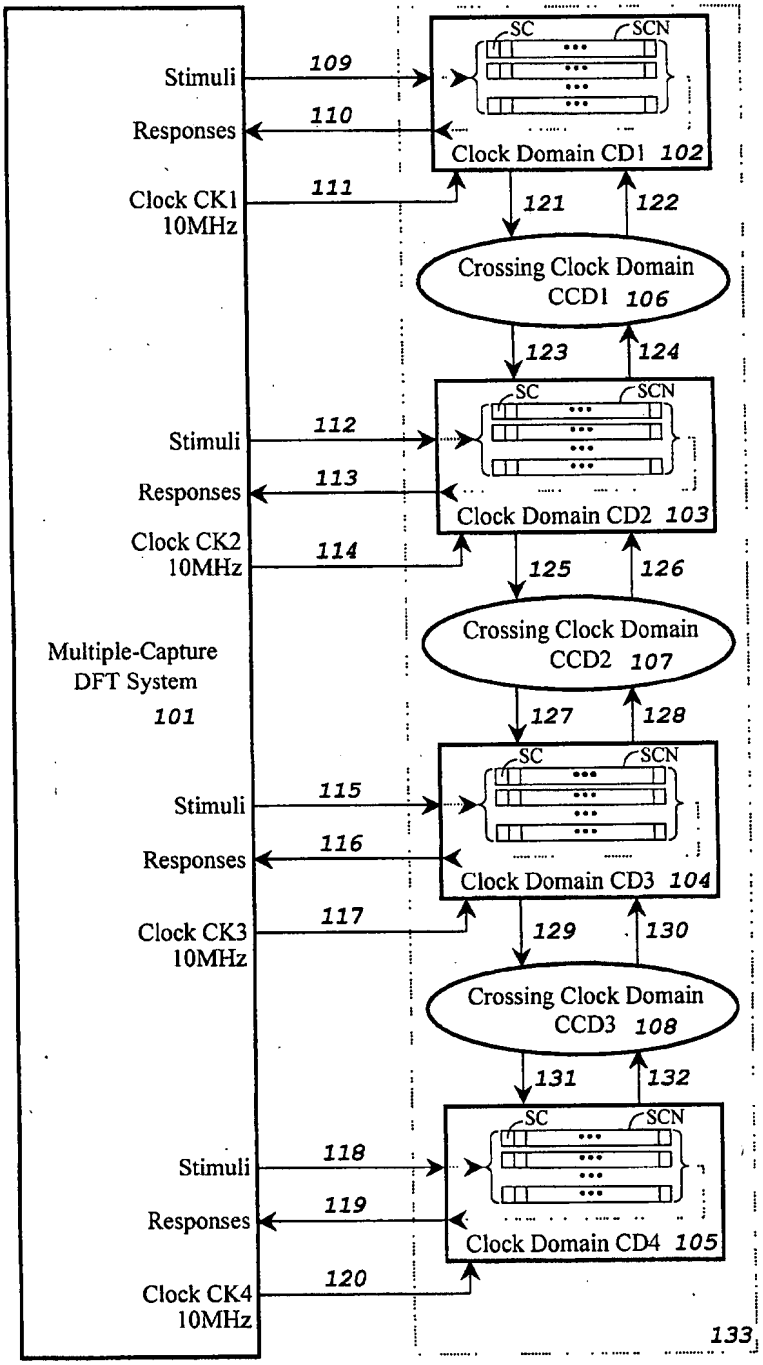

Stimuli
109
Responses
110
Clock CK1
10MHz
111
SC
SCN
Clock Domain CD1 102
121
122
Crossing Clock Domain
CCD1 106
123
124
Stimuli
112
Responses
113
Clock CK2
10MHz
114
Clock Domain CD2 103
125
126
Crossing Clock Domain
CCD2 107
127
128
Multiple-Capture
DFT System
101
Stimuli
115
Responses
116
Clock CK3
10MHz
117
Clock Domain CD3 104
129
130
Crossing Clock Domain
CCD3 108
131
132
Stimuli
118
Responses
119
Clock CK4
10MHz
120
Clock Domain CD4 105
133

FIG. 1

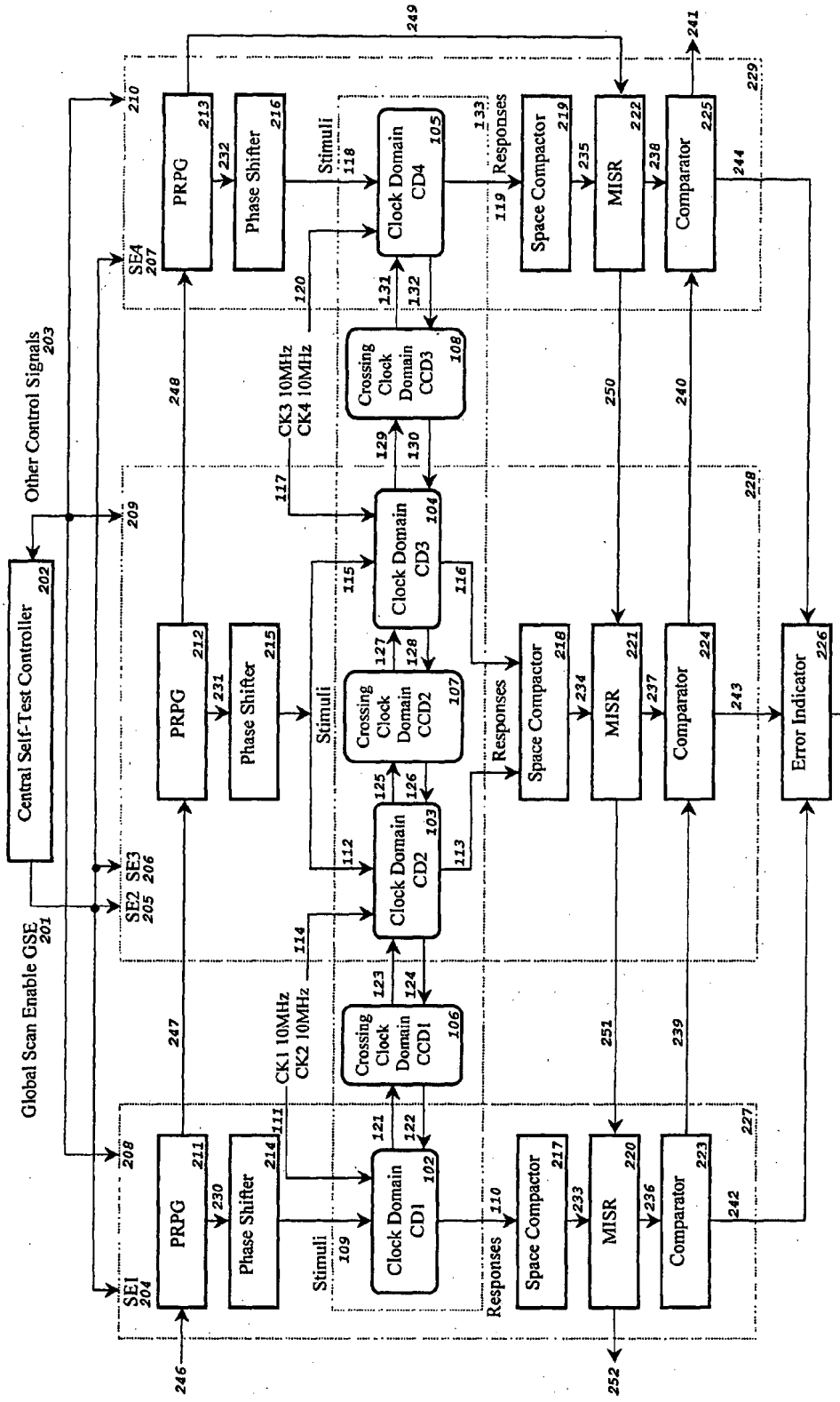

Central Self-Test Controller 202
Other Control Signals 203
Global Scan Enable GSE 201
SE1 204
SE2 205
SE3 206
SE4 207
PRPG 211
Phase Shifter 214
Clock Domain CD1 102
Crossing Clock Domain CCD1 106
Clock Domain CD2 103
Crossing Clock Domain CCD2 107
Clock Domain CD3 104
Crossing Clock Domain CCD3 108
Clock Domain CD4 105
Space Compactor 217
MISR 220
Comparator 223
Error Indicator 226
Pass/Fail 245
CK1 10MHz
CK2 10MHz
CK3 10MHz
CK4 10MHz
Stimuli 109
Responses 110

FIG. 2

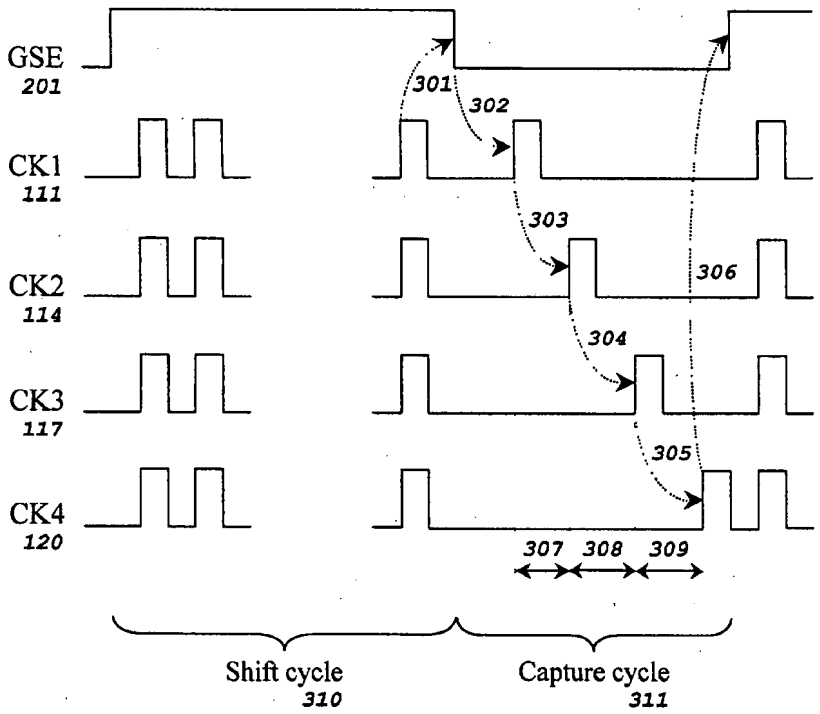

GSE
201
CK1
111
CK2
114
CK3
117
CK4
120
301
302
303
304
305
306
307
308
309
Shift cycle
310
Capture cycle
311

FIG. 3

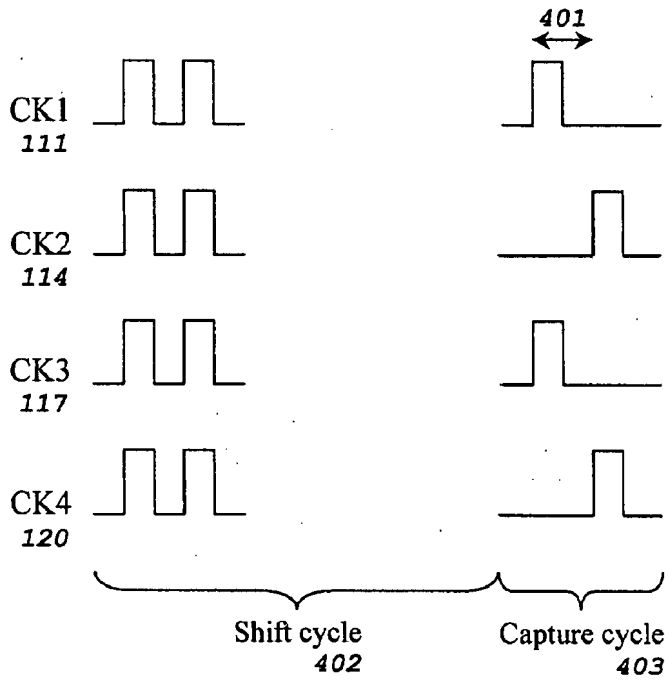

401
CK1
111
CK2
114
CK3
117
CK4
120
Shift cycle
402
Capture cycle
403

FIG. 4

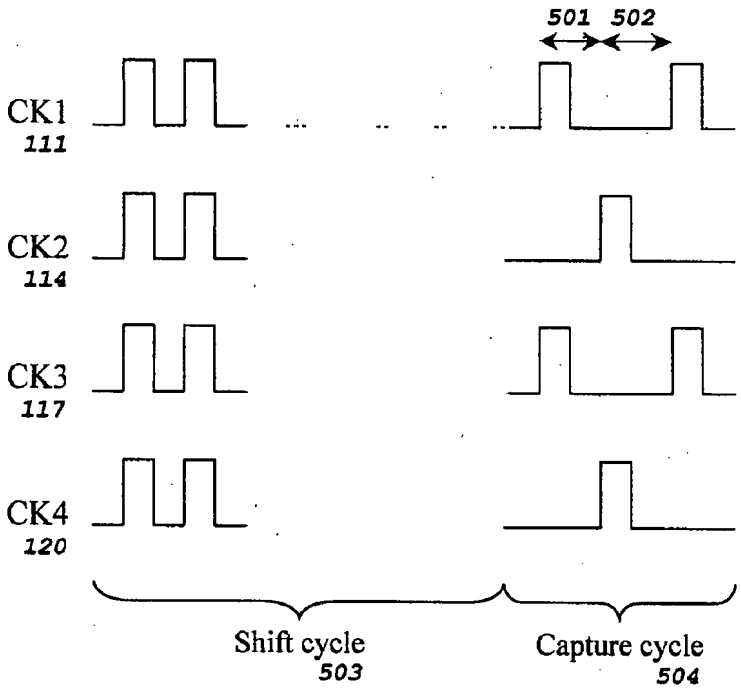

501 502
CK1
111
CK2
114
CK3
117
CK4
120
Shift cycle
503
Capture cycle
504

FIG. 5

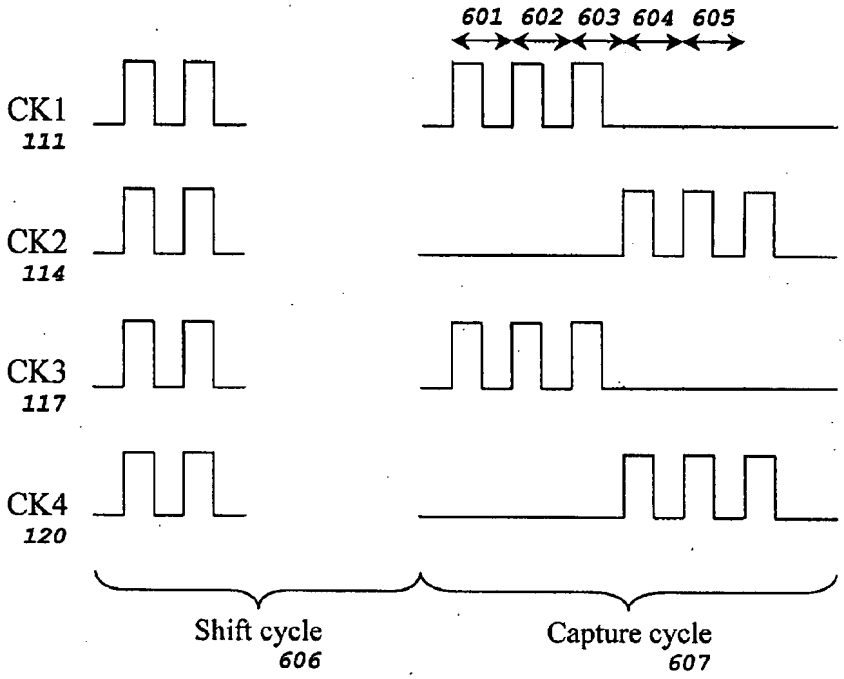

601 602 603 604 605
CK1
111
CK2
114
CK3
117
CK4
120
Shift cycle
606
Capture cycle
607

FIG. 6

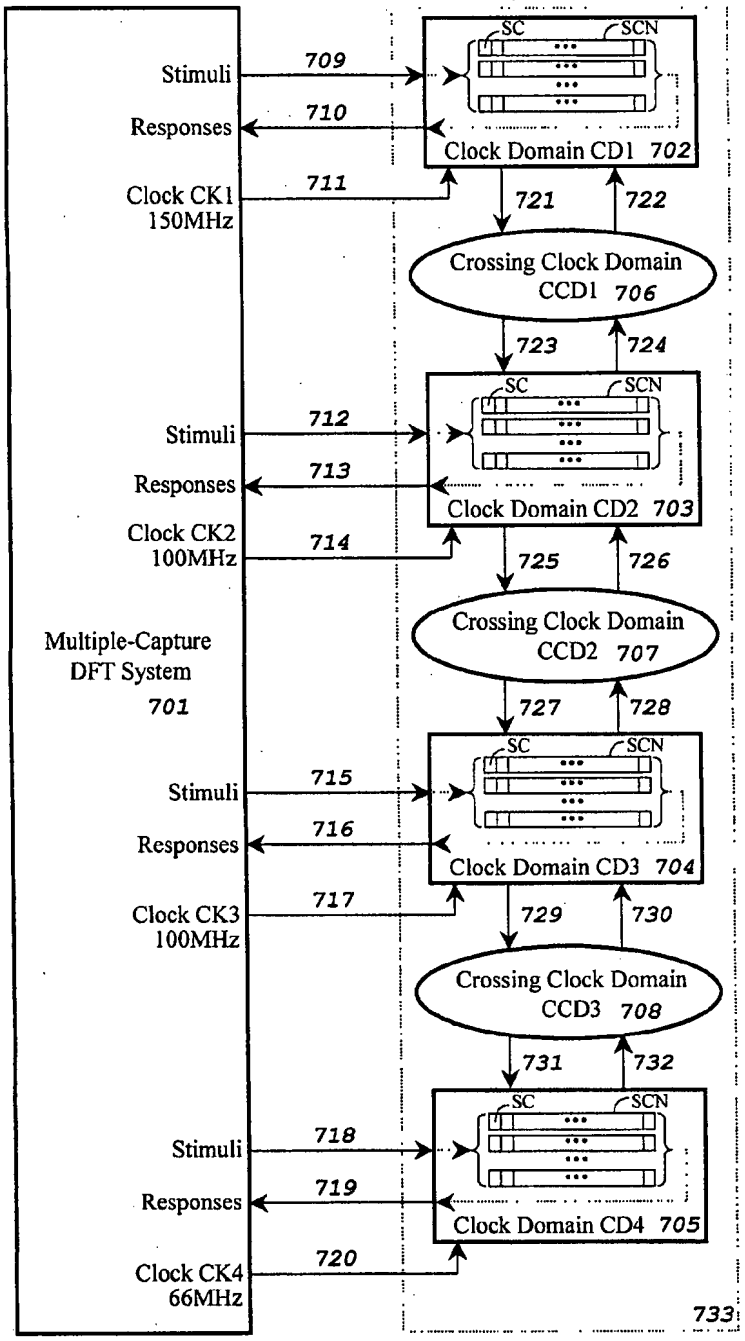

Stimuli
709
Responses
710
Clock CK1
150MHz
711
SC
SCN
Clock Domain CD1 702
721
722
Crossing Clock Domain
CCD1 706
723
724
Stimuli
712
Responses
713
Clock CK2
100MHz
714
Clock Domain CD2 703
725
726
Crossing Clock Domain
CCD2 707
727
728
Multiple-Capture
DFT System
701
Stimuli
715
Responses
716
Clock CK3
100MHz
717
Clock Domain CD3 704
729
730
Crossing Clock Domain
CCD3 708
731
732
Stimuli
718
Responses
719
Clock CK4
66MHz
720
Clock Domain CD4 705
733

FIG. 7

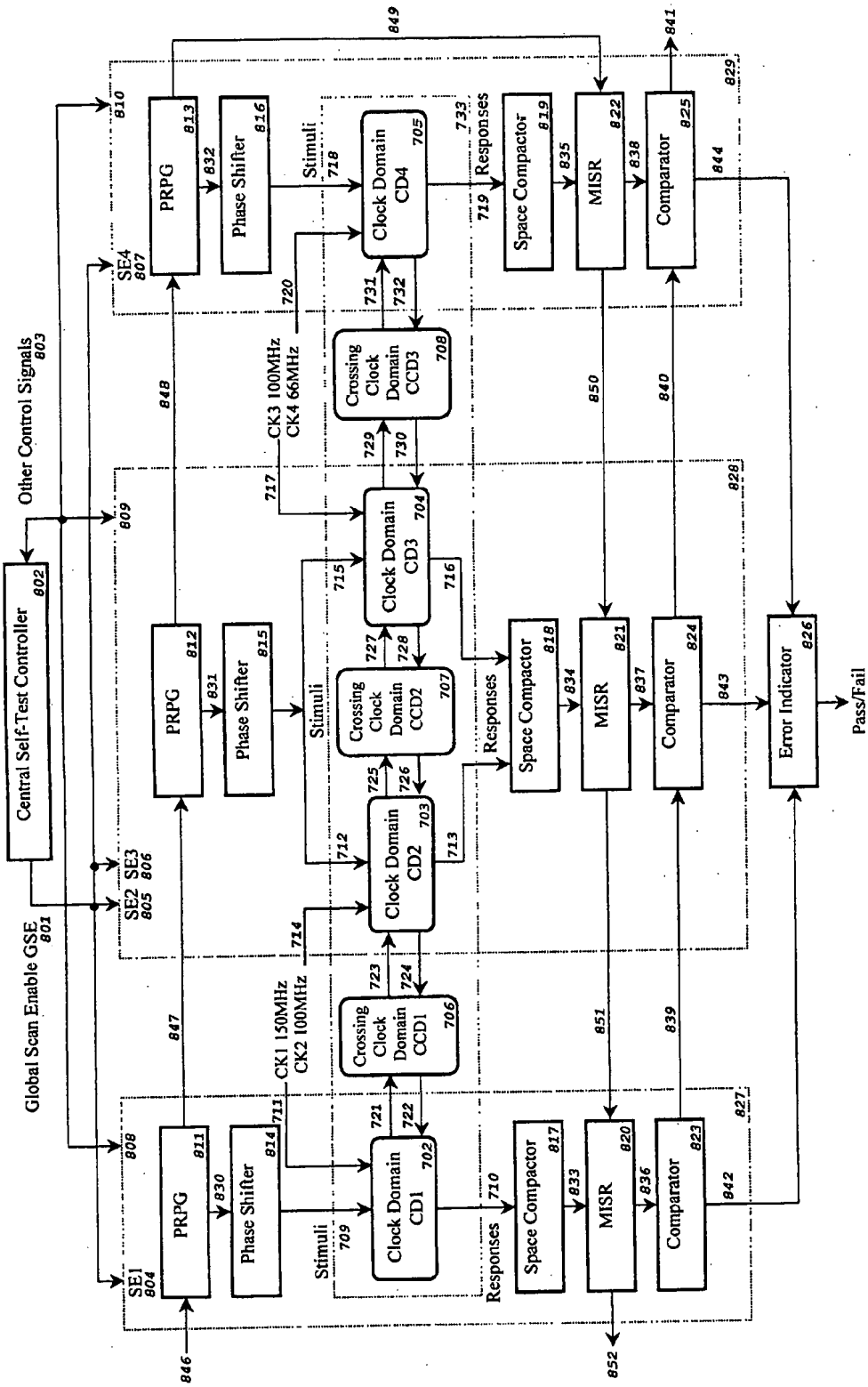

Central Self-Test Controller 802
Other Control Signals 803
Global Scan Enable GSE 801
SE1 804
SE2 805
SE3 806
SE4 807
PRPG 811
Phase Shifter 814
Clock Domain CD1 702
Crossing Clock Domain CCD1 706
Clock Domain CD2 703
Crossing Clock Domain CCD2 707
Clock Domain CD3 704
Crossing Clock Domain CCD3 708
Clock Domain CD4 705
Space Compactor 817
MISR 820
Comparator 823
Error Indicator 826
Pass/Fail 845
CK1 150MHz CK2 100MHz
CK3 100MHz CK4 66MHz
Stimuli 709
Responses 710

FIG. 8

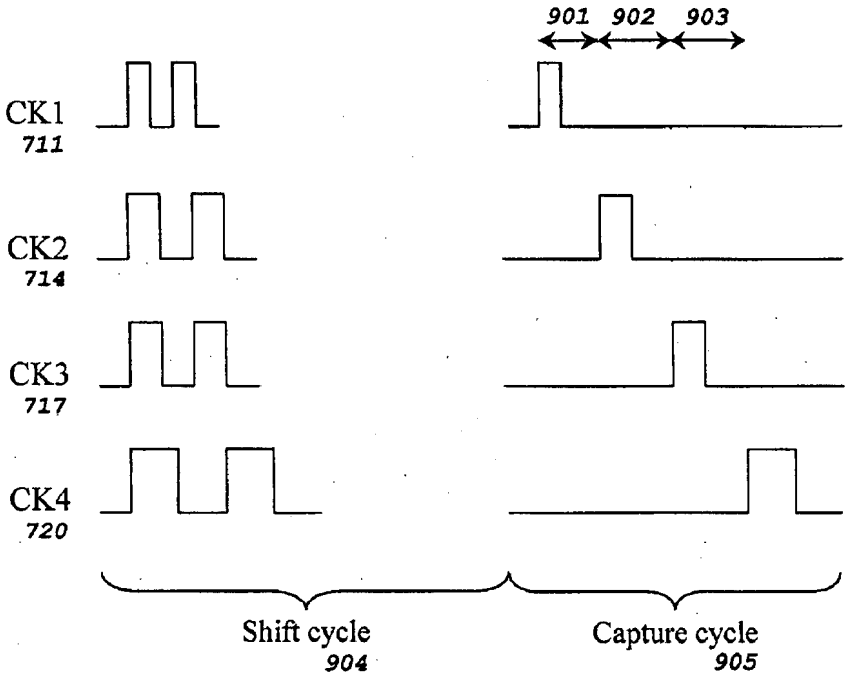

901
902
903
CK1
711
CK2
714
CK3
717
CK4
720
Shift cycle
904
Capture cycle
905

FIG. 9

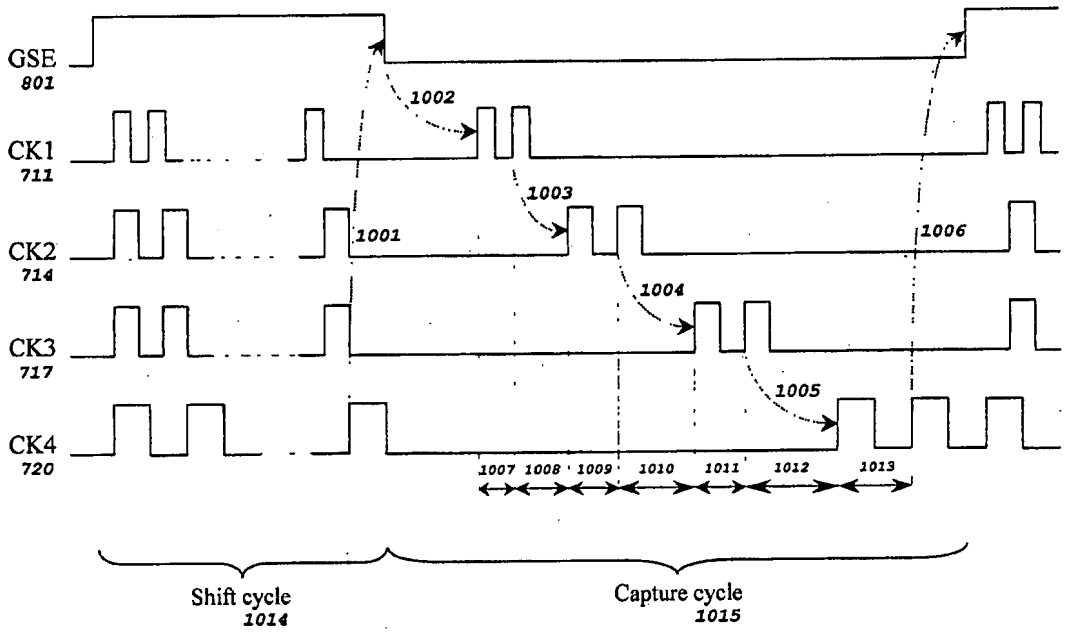

GSE
801
CK1
711
CK2
714
CK3
717
CK4
720
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
Shift cycle
1014
Capture cycle
1015

FIG. 10

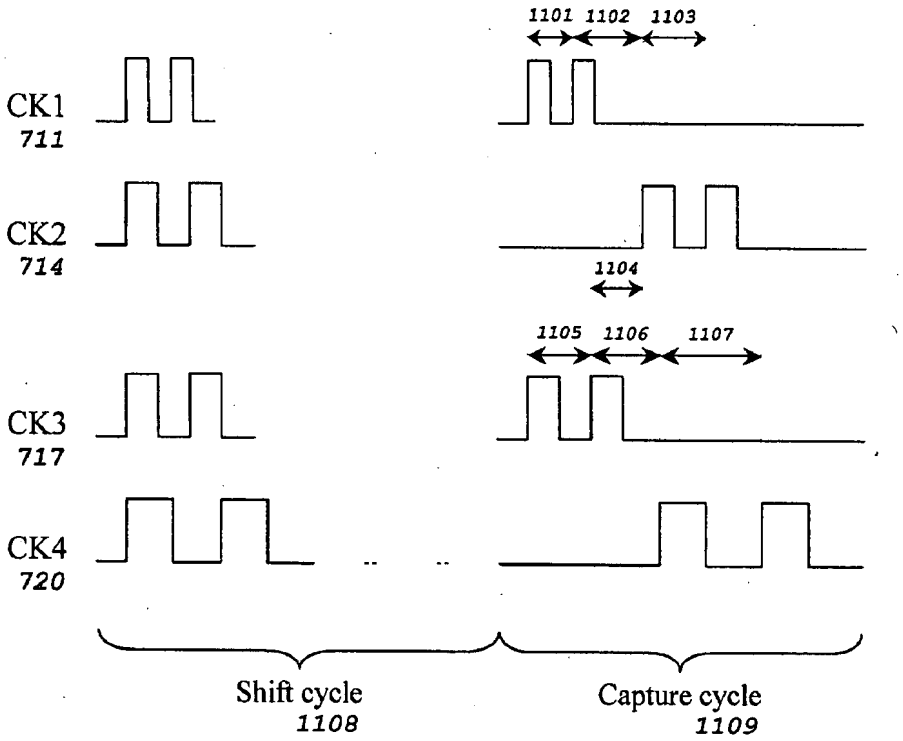

1101 1102 1103
CK1
711
CK2
714
1104
1105 1106 1107
CK3
717
CK4
720
Shift cycle
1108
Capture cycle
1109

FIG. 11

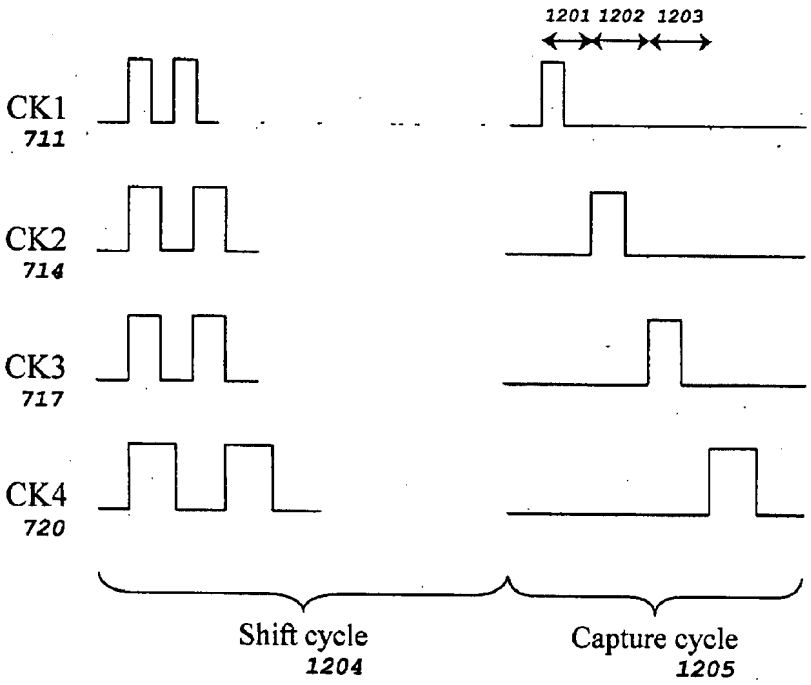

1201 1202 1203
CK1
711
CK2
714
CK3
717
CK4
720
Shift cycle
1204
Capture cycle
1205

FIG. 12

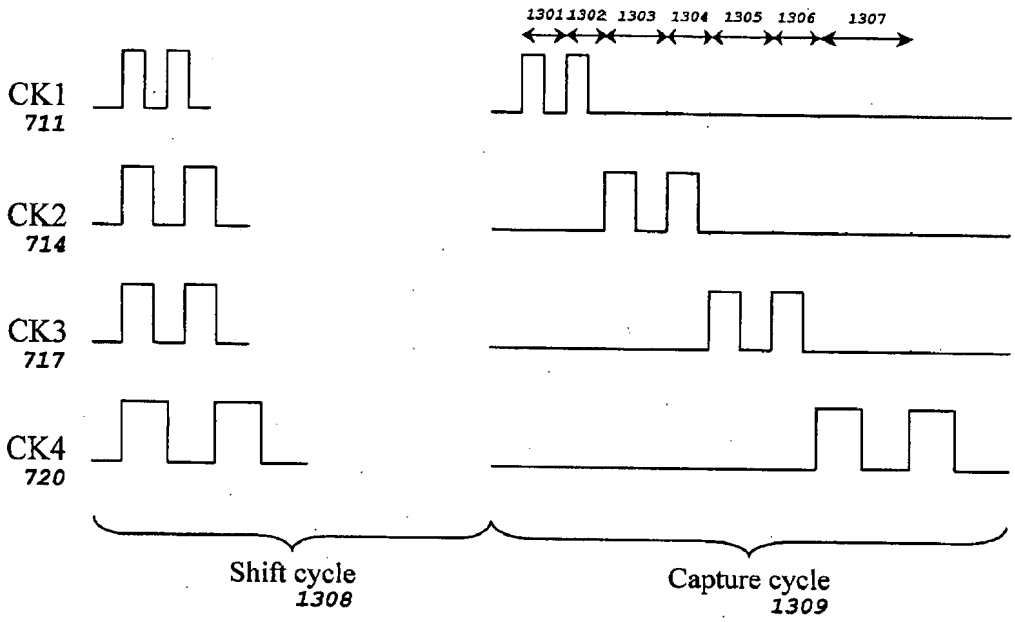

1301 1302 1303 1304 1305 1306 1307
CK1
711
CK2
714
CK3
717
CK4
720
Shift cycle
1308
Capture cycle
1309

FIG. 13

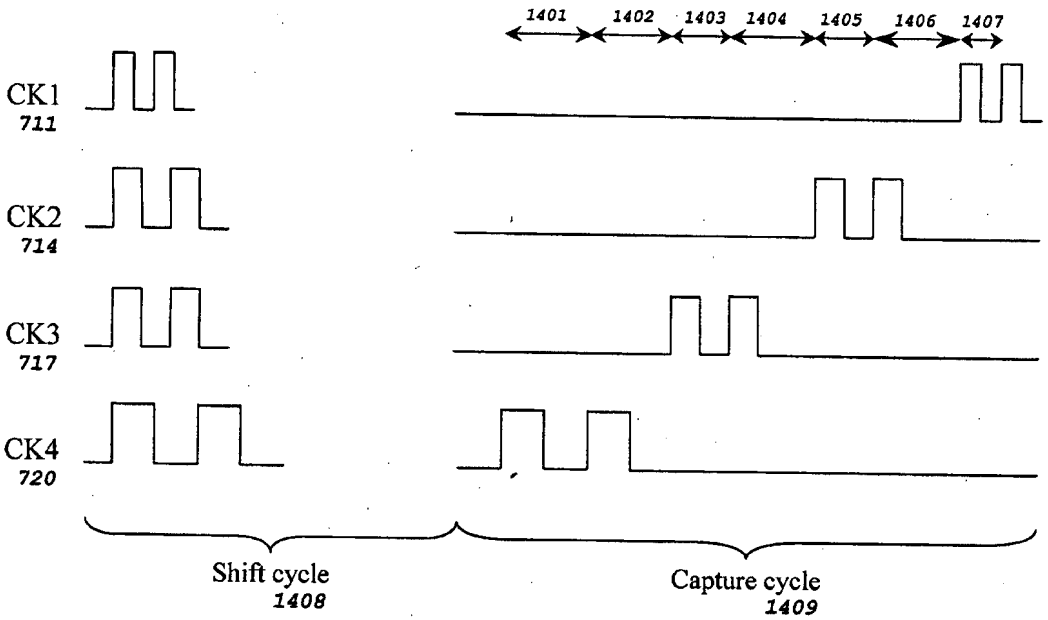

1401
1402
1403
1404
1405
1406
1407
CK1
711
CK2
714
CK3
717
CK4
720
Shift cycle
1408
Capture cycle
1409

FIG. 14

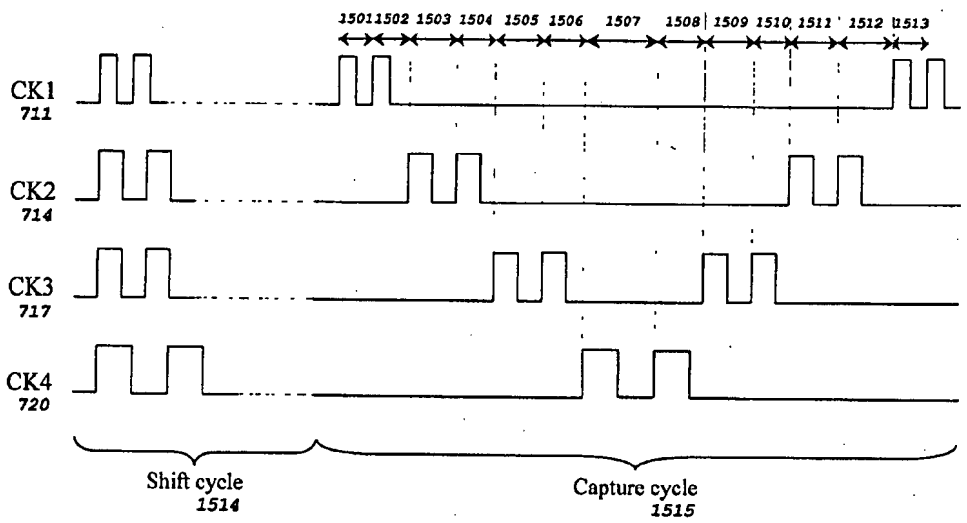

15011502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513
CK1
711
CK2
714
CK3
717
CK4
720
Shift cycle
1514
Capture cycle
1515

FIG. 15

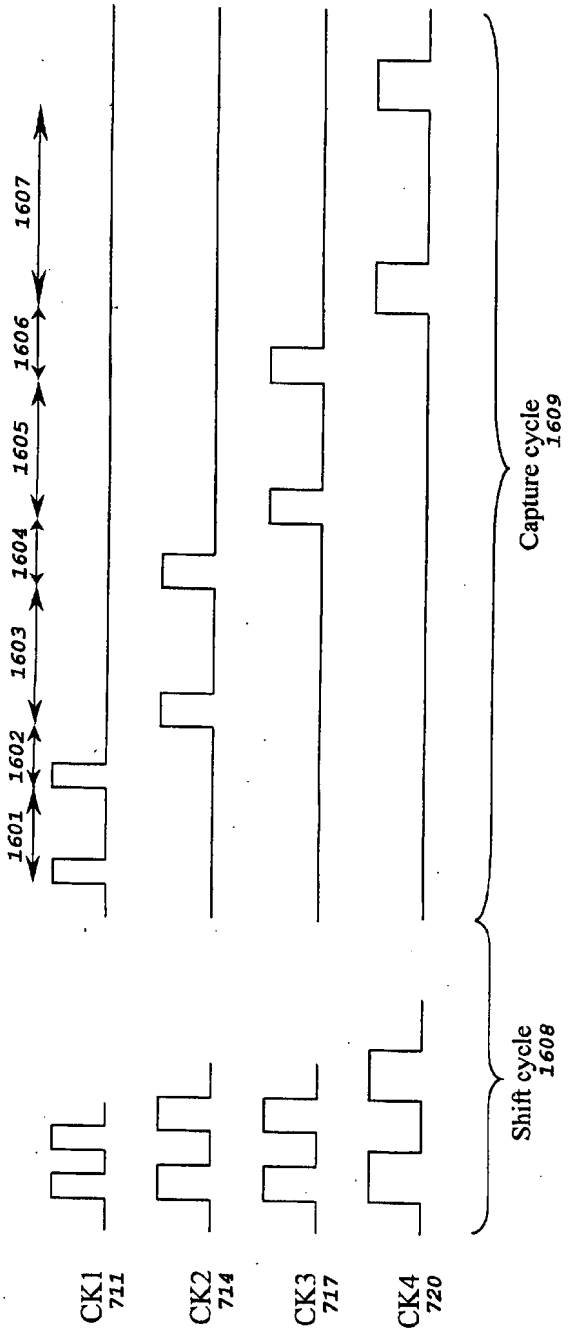

1601
1602
1603
1604
1605
1606
1607
Capture cycle
1609
Shift cycle
1608
CK1
711
CK2
714
CK3
717
CK4
720

FIG. 16

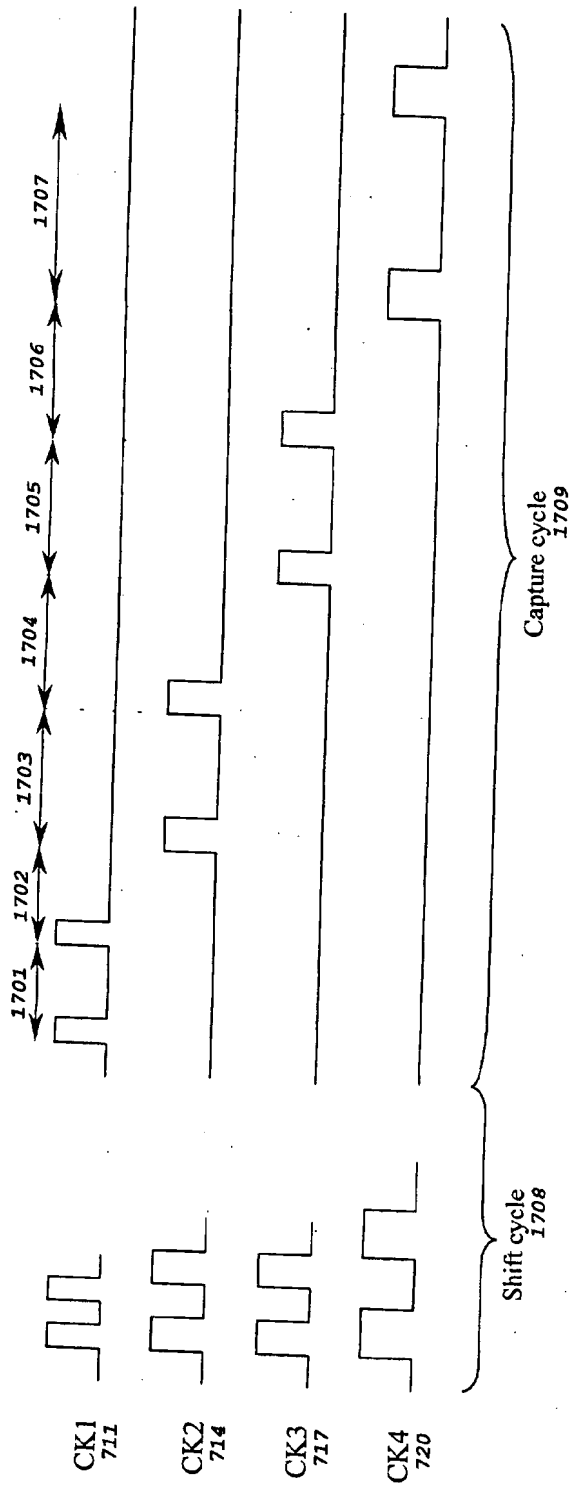

1701
1702
1703
1704
1705
1706
1707
Capture cycle
1709
Shift cycle
1708
CK1
711
CK2
714
CK3
717
CK4
720

FIG. 17

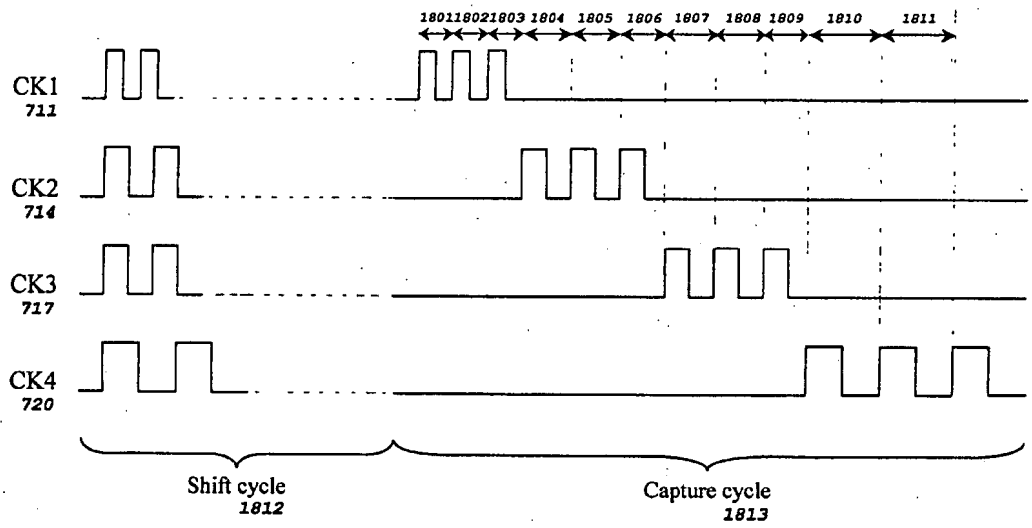

1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811
CK1
711
CK2
714
CK3
717
CK4
720
Shift cycle
1812
Capture cycle
1813

FIG. 18

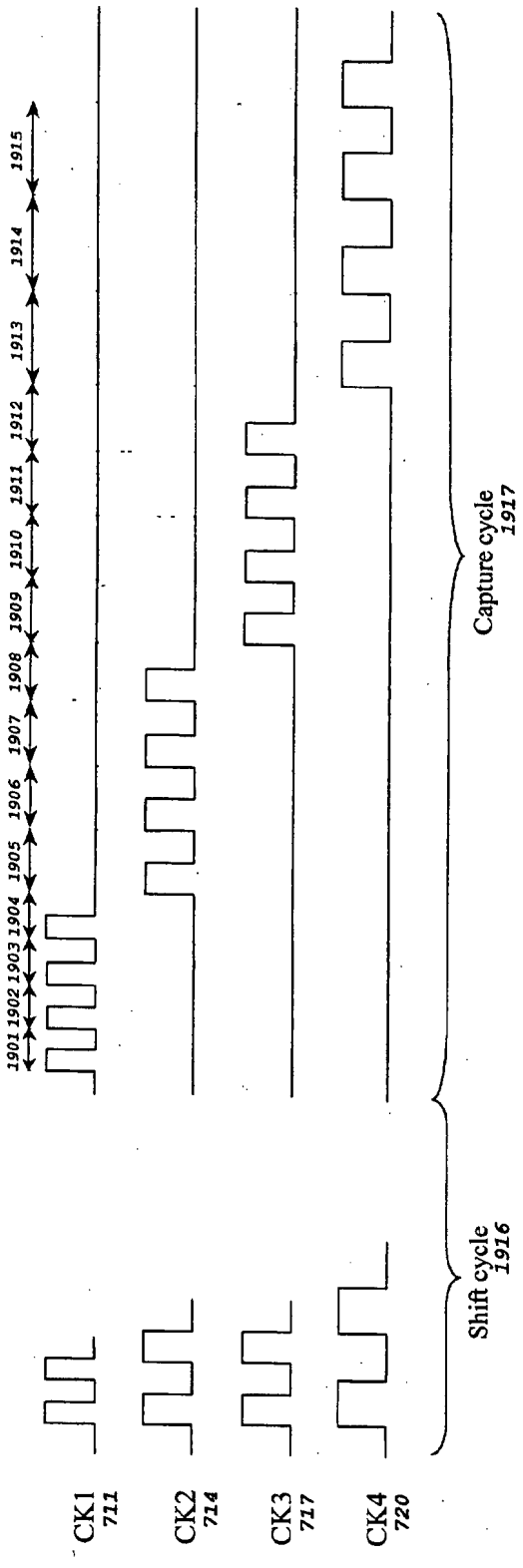

1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
CK1
711
CK2
714
CK3
717
CK4
720
Shift cycle
1916
Capture cycle
1917

FIG. 19

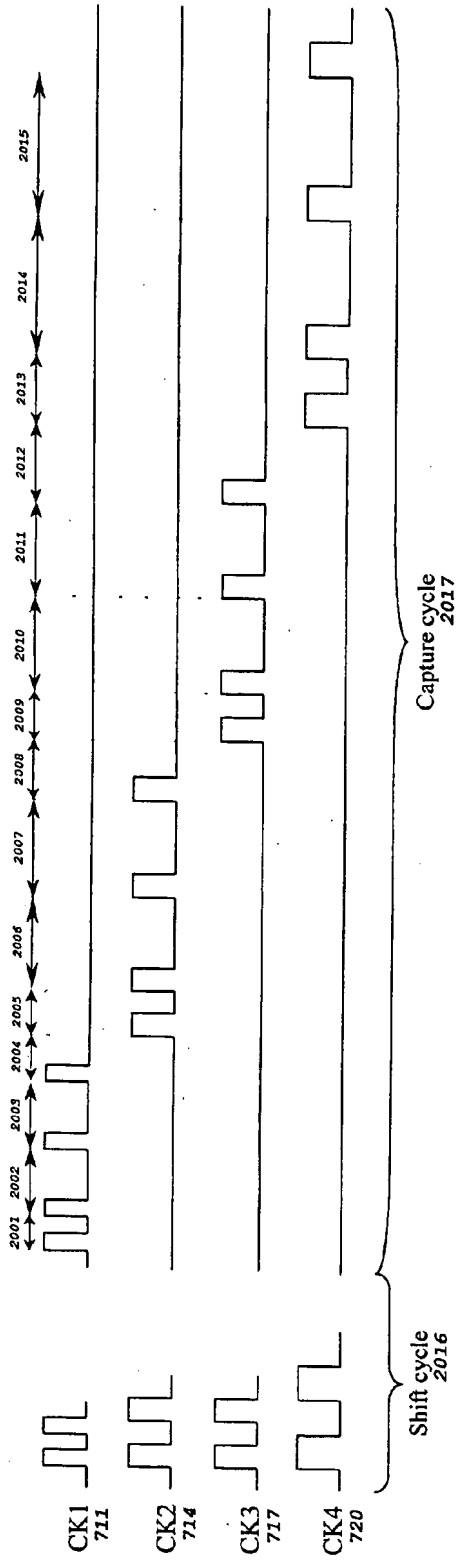

2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
CK1
711
CK2
714
CK3
717
CK4
720
Shift cycle
2016
Capture cycle
2017

FIG. 20

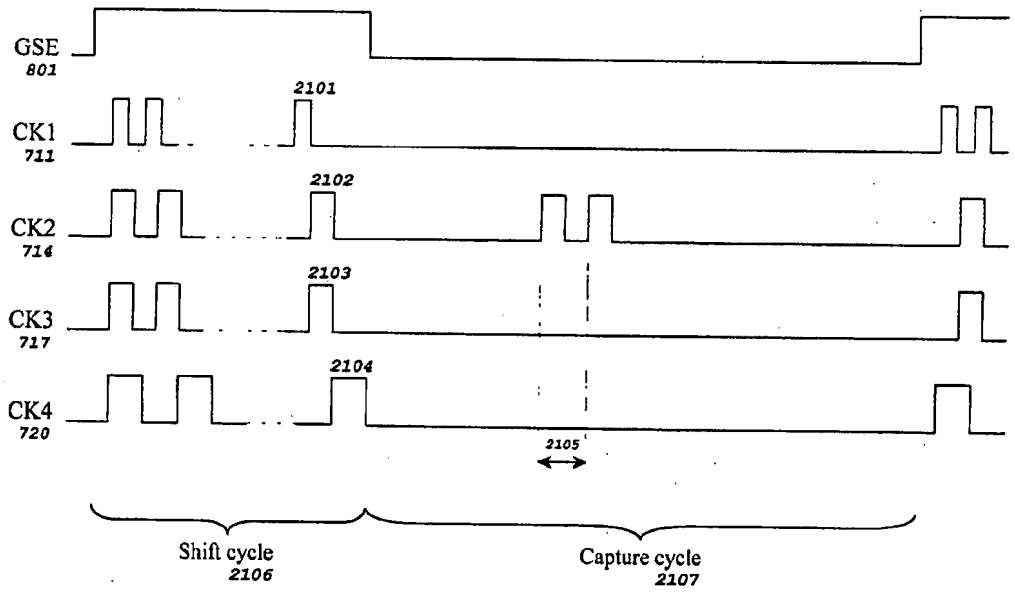

GSE
801
2101
CK1
711
2102
CK2
714
2103
CK3
717
2104
CK4
720
2105
Shift cycle
2106
Capture cycle
2107

FIG. 21

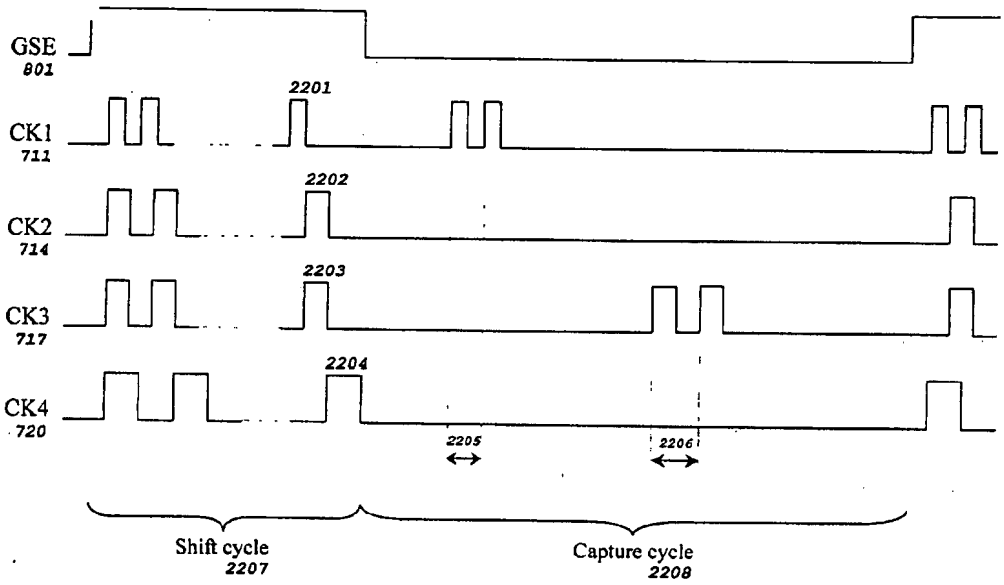

GSE
801
2201
CK1
711
2202
CK2
714
2203
CK3
717
2204
CK4
720
2205
2206
Shift cycle
2207
Capture cycle
2208

FIG. 22

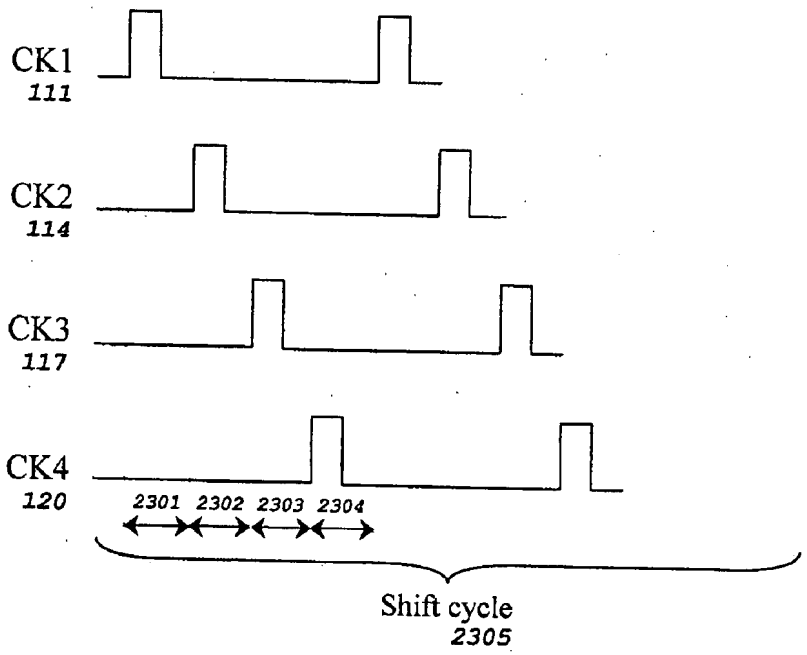

CK1
111
CK2
114
CK3
117
CK4
120
2301 2302 2303 2304
Shift cycle
2305

FIG. 23

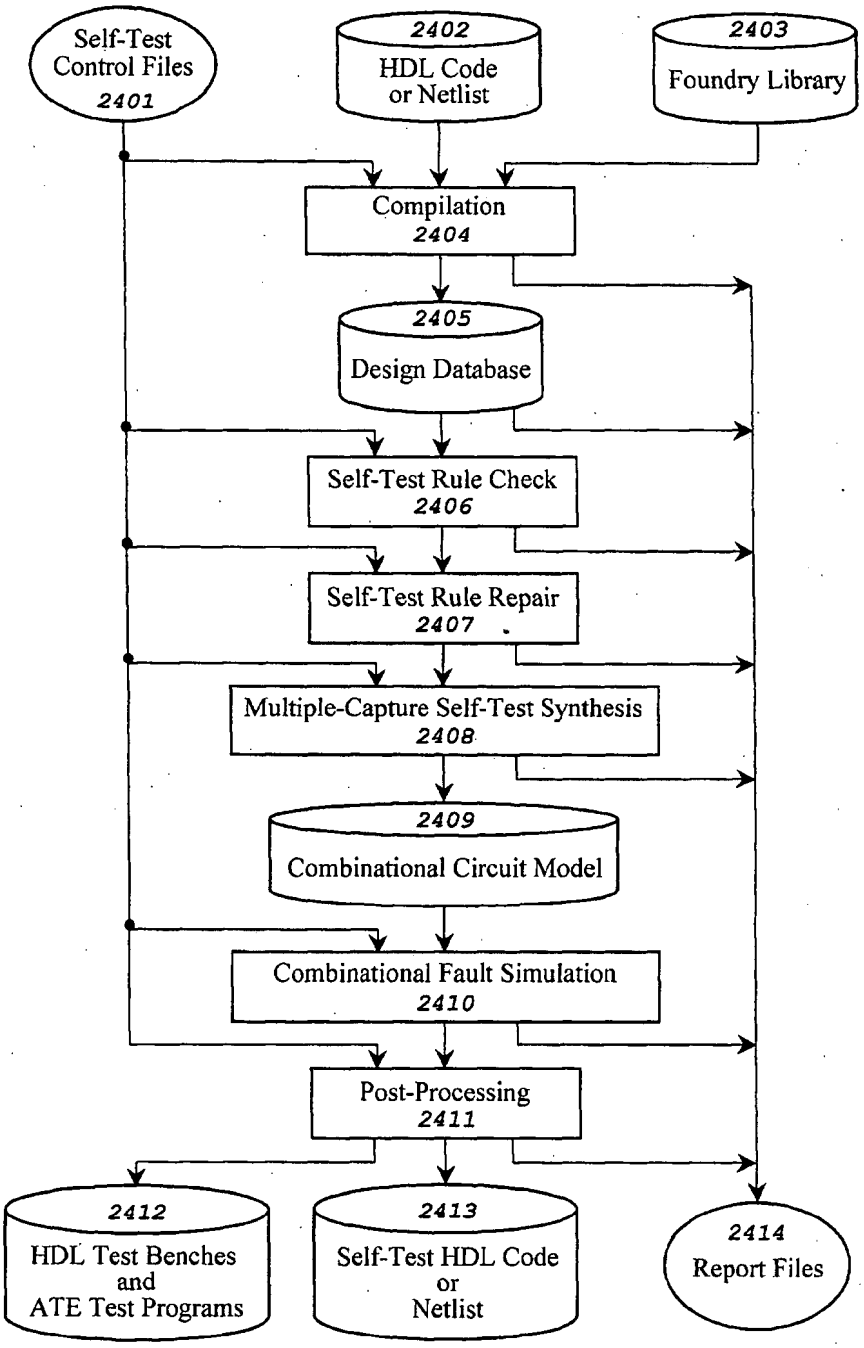

Self-Test Control Files 2401
2402 HDL Code or Netlist
2403 Foundry Library
Compilation 2404
2405 Design Database
Self-Test Rule Check 2406
Self-Test Rule Repair 2407
Multiple-Capture Self-Test Synthesis 2408
2409 Combinational Circuit Model
Combinational Fault Simulation 2410
Post-Processing 2411
2412 HDL Test Benches and ATE Test Programs
2413 Self-Test HDL Code or Netlist
2414 Report Files

FIG. 24

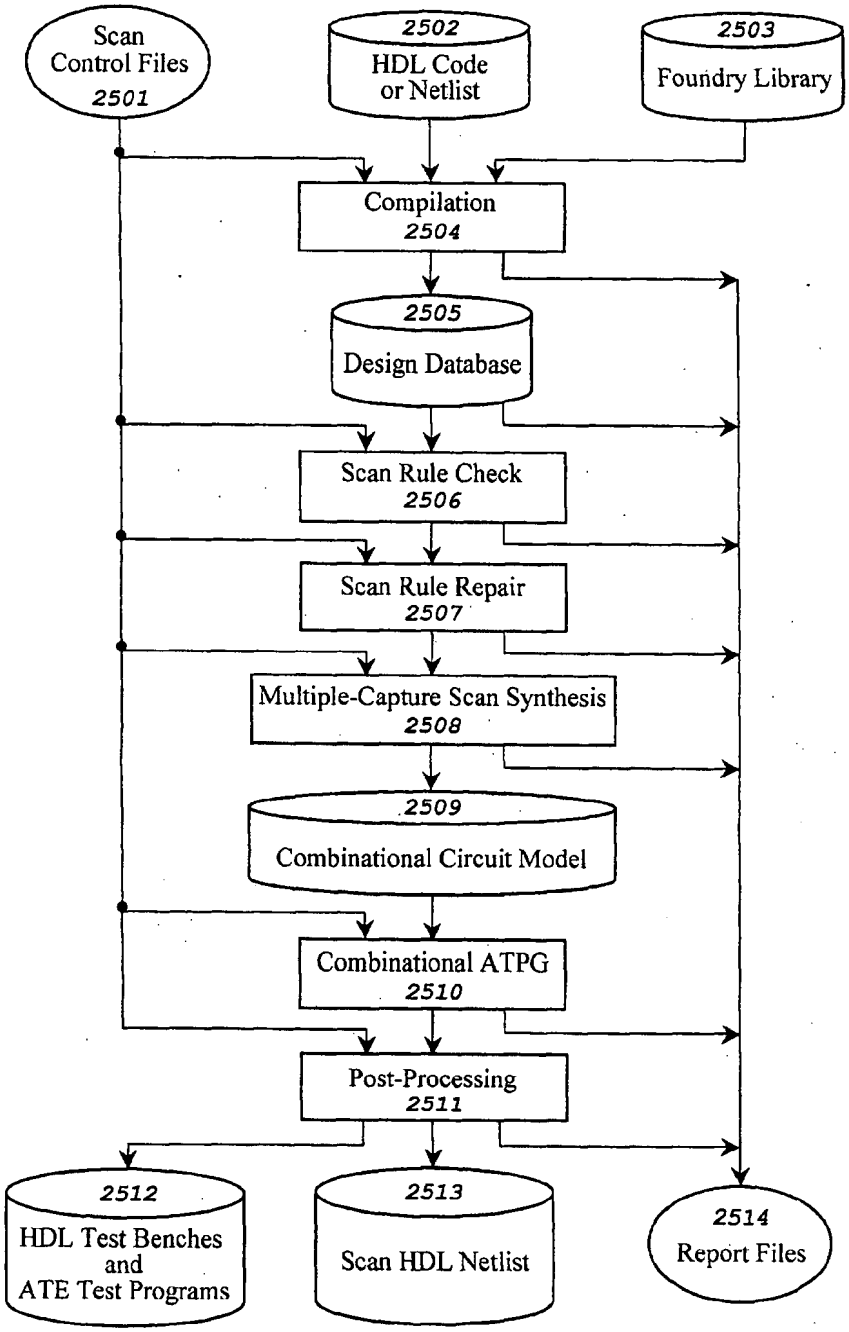

Scan Control Files 2501
2502 HDL Code or Netlist
2503 Foundry Library
Compilation 2504
2505 Design Database
Scan Rule Check 2506
Scan Rule Repair 2507
Multiple-Capture Scan Synthesis 2508
2509 Combinational Circuit Model
Combinational ATPG 2510
Post-Processing 2511
2512 HDL Test Benches and ATE Test Programs
2513 Scan HDL Netlist
2514 Report Files

FIG. 25

MULTIPLE-CAPTURE DFT SYSTEM FOR DETECTING OR LOCATING CROSSING CLOCK-DOMAIN FAULTS DURING SELF-TEST OR SCAN-TEST

RELATED APPLICATION DATA

[0001] This application claims the benefit of U.S. Provisional Application No. 60/268,601 filed Feb. 15, 2001, which is hereby incorporated by reference.

TECHNICAL FIELD

[0002] The present invention generally relates to the testing of logic designs in an integrated circuit or circuit assembly embedded with design-for-test (DFT) techniques. Specifically, the present invention relates to the detection or location of logic faults within each clock domain and logic faults crossing any two clock domains, during self-test or scan-test, in an integrated circuit or circuit assembly.

BACKGROUND OF THE INVENTION

[0003] In this specification, the term integrated circuit is used to describe a chip or MCM (multi-chip module) embedded with design-for-test (DFT) techniques. The terms circuit assembly and printed circuit board will be considered interchangeable. The term circuit assembly includes printed circuit boards as well as other types of circuit assemblies. A circuit assembly is a combination of integrated circuits. The resulting combination is manufactured to form a physical or functional unit.

[0004] An integrated circuit or circuit assembly, in general, contains two or more systems clocks, each controlling one module or logic block, called clock domain. Each system clock is either directly coming from a primary input (edge pin/connector) or generated internally. These system clocks can operate at totally unrelated frequencies (clock speeds), at sub-multiples of each other, at the same frequency but with different clock skews, or at a mix of the above. Due to clock skews among these system clocks, when a DFT technique, such as self-test or scan-test, is employed, it is very likely that faults associated with the function between two clock domains, called crossing clock-domain faults, will become difficult to test. In the worst case, these crossing clock-domain faults when propagating into the receiving clock domain could completely block detection or location of all faults within that clock domain. Thus, in order to solve the fault propagation problem, DFT approaches are proposed to take over control of all system clocks and reconfigure them as capture clocks.

[0005] Prior-art DFT approaches in this area to testing crossing clock-domain faults as well as faults within each clock domain centered on using the isolated DFT, ratio'ed DFT, and one-hot DFT techniques. They are all referred to as single-capture DFT techniques, because none of them can provide multiple skewed capture clocks (or an ordered sequence of capture clocks) in each capture cycle during self-test or scan-test.

[0006] In using the isolated DFT technique, all boundary signals crossing a clock domain and flowing into the receiving clock domains are completely blocked or disabled by forcing each of them to a predetermined logic value of 0 or 1. See U.S. Pat. No. 6,327,684 issued to Nadeau-Dostie et al. (2001). This approach, in general, can allow all clock domains to be tested in parallel. The major drawbacks of this approach are that it requires insertion of capture-disabled logic in between clock domains and all scan enable signals each associated with one clock domain must be operated at-speed. The design change could take significant efforts and it might impact normal mode operation. Running all scan enable signals at-speed requires routing them as clock signals using layout clock-tree synthesis (CTS). In addition, since boundary signals can traverse through two clock domains in both directions, this approach requires testing crossing clock-domain faults in two or more test sessions. This could substantially increase the test time required and might make the capture-disabled logic even more complex to implement than anticipated.

[0007] In using the ratio'ed DFT technique, all clock domains must be operated at sub-multiples of one reference clock. For instance, assume that a design contains 3 clock domains running at 150 MHz, 80 MHz, and 45 MHz, respectively. The 3 clock domains may have to be operated at 150 MHz, 75 MHz, and 37.5 MHz during testing. See U.S. Pat. No. 5,349,587 issued to Nadeau-Dostie et al. (1994). This approach reduces the complexity of testing a multiple-frequency design and avoids potential races or timing violations crossing clock domains. It can also allow testing of all clock domains in parallel. However, due to changes in clock-domain operating frequencies, this approach loses its self-test or scan-test intent of testing multiple-frequency designs at their rated clock speeds (at-speed) and may require significant design and layout efforts on re-timing (or synchronizing) all clock domains. Power consumption could be also another serious problem because all scan cells (memory elements) are triggered simultaneously every few cycles.

[0008] In using the one-shot DFT technique, each crossing clock-domain signal flowing into its receiving clock domains must be initialized to or held at a predetermined logic value of 0 or 1 first. This initialization is usually accomplished by shifting in predetermined logic values to all clock domains so that all crossing clock-domain signals are forced to a known state. Testing is then conducted domain-by-domain, thus, called one-hot testing. See U.S. Pat. No. 5,680,543 issued to Bhawmik et al. (1997). The major benefits of using this approach are that it can still detect or locate crossing clock-domain faults and does not need insertion of disabled logic, in particular, in critical paths crossing clock domains. However, unlike the isolated or ratio'ed DFT approach, this approach requires testing of all clock domains in series, resulting in long test time. It also requires significant design and layout efforts on re-timing (or synchronizing) all clock domains.

[0009] Two additional prior-art DFT approaches had also been proposed, one for scan-test, the other for self-test. Both approaches are referred to as multiple-capture DFT techniques, because they can provide multiple skewed capture clocks (or an ordered sequence of capture clocks) in each capture cycle during scan-test or self-test.

[0010] The first prior-art multiple-capture DFT approach is to test faults within each clock domain and faults between two clock domains in scan-test mode. See U.S. Pat. No. 6,070,260 issued to Buch et al. (2000) and U.S. Pat. No. 6,195,776 issued to Ruiz et al. (2001). These approaches rest

on using multiple skewed scan clocks or multiple skew capture events each operating at the same reduced clock speed in an ATE (automatic test equipment) to detect faults. Combinational ATPG (automatic test pattern generation) is used to generate scan-test patterns and ATE test programs are created to detect faults in the integrated circuit. Unfortunately, currently available ATPG tools only assume the application of one clock pulse (clock cycle) to each clock domain. Thus, these approaches can only detect stuck-at faults in scan-test mode. No prior art using multiple skewed capture clocks were proposed to test delay or stuck-at faults requiring two or more capture clock pulses for full-scan or partial-scan designs.

[0011] The second prior-art multiple-capture DFT approach is to test faults within each clock domain and faults between two clock domains in self-test mode. See the paper co-authored by Hetherington et al. (1999). This approach rests on using multiple shift-followed-by-capture clocks each operating at its operating frequency, in a programmable capture window, to detect faults at-speed. It requires clock suppression, complex scan enable (SE) timing waveforms, and shift clock pulses in the capture window to control the capture operation. These shift clock pulses may also need precise timing alignment. As a result, it becomes quite difficult to perform at-speed self-test for designs containing clock domains operated at totally unrelated frequencies, e.g., 133 MHz and 60 MHz.

[0012] Thus, there is a need for an improved method, apparatus, or computer-aided design (CAD) system that allows at-speed or slow-speed testing of faults within clock domains and between any two clock domains using a simple multiple-capture DFT technique. The method and apparatus of the present invention will control the multiple-capture operations of the capture clocks in self-test or scan-test mode. It does not require using shift clock pulses in the capture window, inserting capture-disabled logic in normal mode, applying clock suppression on capture clock pulses, and programming complex timing waveforms on scan enable (SE) signals. In addition, the CAD system of the present invention further comprises the computer-implemented steps of performing multiple-capture self-test or scan synthesis, combinational fault simulation, and combinational ATPG that are currently unavailable in the CAD field using multiple-capture DFT techniques.

SUMMARY OF THE INVENTION

[0013] Accordingly, a primary objective of the present invention is to provide an improved multiple-capture DFT system implementing the multiple-capture DFT technique. Such a DFT system will comprise a method or apparatus for allowing at-speed/slow-speed detection or location of faults within all clock domains and faults crossing clock domains in an integrated circuit or circuit assembly. In the present invention, the method or apparatus can be realized and placed inside or external to the integrated circuit or circuit assembly.

[0014] A computer-aided design (CAD) system that synthesizes such a DFT system and generates desired HDL test benches and ATE test programs is also included in the present invention. A hardware description language (HDL) is used to represent the integrated circuit includes, but is not limited to, Verilog or VHDL. An ATE is an IC tester or any equipment that realizes the multiple-capture DFT system and is external to the integrated circuit or circuit assembly under test.

[0015] The present invention focuses on multiple-capture DFT systems for self-test and scan-test. In a self-test environment, a self-test cycle often comprises 3 major operations: shift, capture, and compact. The shift and compact operations can occur concurrently during each self-test cycle. In order to increase the circuit's fault coverage, it is often necessary to include scan-test cycles to perform top-up ATPG. A scan-test cycle often comprises 3 major operations in a scan-test environment: shift, capture, and compare. The shift and compare operations can occur concurrently during each scan-test cycle. In a mixed self-test and scan-test environment, the scan-test cycle may execute a compact operation rather than the compare operation. Thus, in the present invention, a self-test cycle further comprises the shift, capture, and compare operations, and a scan-test cycle further comprises the shift, capture, and compact operations.

[0016] The multiple-capture DFT system of the present invention further comprises any method or apparatus for executing the shift and compact or shift and compare operations concurrently during each self-test or scan-test cycle. It is applicable to test any integrated circuit or circuit assembly which contains N clock domains, where N>1. Each capture clock controls one clock domain and can operate at its rated clock speed (at-speed) or at a reduced clock speed (slow-speed), when desired.

[0017] During the shift operation, the multiple-capture DFT system first generates and shifts in (loads) N pseudo-random or predetermined stimuli to all scan cells within all clock domains, concurrently. The shifting frequency is irrelevant to at-speed testing. Depending on needs, a slower frequency can be used to reduce power consumption and a faster frequency can be used to reduce the test application time. The multiple-capture DFT system must wait until all stimuli have been loaded or shifted into all scan cells. By that time, all scan enable (SE) signals each associated with one clock domain shall switch from the shift operation to the capture operation. After the capture operation is completed, all scan enable (SE) signals shall switch from the capture operation to the shift operation. One global scan enable (GSE) signal can be simply used to drive these scan enable signals.

[0018] The multiple-capture DFT system of the present invention further comprises any method or apparatus for performing the shift operation at any selected clock speed within each clock domain and using only one global scan enable (GSE) signal to drive all scan enable (SE) signals for at-speed or slow-speed testing. The GSE signal can be also operated at its selected reduced clock speed. Thus, there is no need to route these SE signals as clock signals using layout clock tree synthesis (CTS). This invention applies to any self-test or scan-test method that requires multiple capture clock pulses (without including shift clock pulses) in the capture cycle.

[0019] After the shift operation is completed, an ordered sequence of capture clocks is applied to all clock domains. During the capture operation, each ordered sequence contains N capture clocks of which only one or a few will be active at one time. There are no shift clock pulses present within each capture cycle. Testing of delay faults at-speed is

now performed by applying two consecutive capture clock pulses (double captures) rather than using the shift-followed-by-capture clock pulses. Performing multiple captures in the capture cycle reduces the risk of delay test invalidation and false paths that might occur due to illegal states in scan cells resulting from filling them with pseudo-random or predetermined stimuli.

[0020] In the present invention, the multiple-capture DFT system uses a daisy-chain clock-triggering or token-ring clock-enabling technique to generate and order capture clocks one after the other. One major benefit of using this approach is that the test results are repeatable no matter what clock speed will be used for each capture clock. The problem is it could be difficult to precisely control the relative clock delay between two adjacent capture clocks for testing delay faults between clock domains.

[0021] As an example, assume that the capture cycle contains 4 capture clocks, CK1, CK2, CK3, and CK4. (Please refer to **FIGS. 3 and 10** in the DETAILED DESCRIPTION OF THE DRAWINGS section for further descriptions). The daisy-chain clock-triggering technique implies that completion of the shift cycle triggers the GSE signal to switch from shift to capture cycle which in turn triggers CK1, the rising edge of the last CK1 pulse triggers CK2, the rising edge of the last CK2 pulse triggers CK3, and the rising edge of the last CK3 pulse triggers CK4. Finally, the rising edge of the last CK4 pulse triggers the GSE signal to switch from capture to shift cycle.

[0022] The token-ring clock-enabling technique implies that completion of the shift cycle enables the GSE signal to switch from shift to capture cycle which in turn enables CK1, completion of CK1 pulses enables CK2, completion of CK2 pulses enables CK3, and completion of CK3 pulses enables CK4. Finally, completion of CK4 pulses enables the GSE signal to switch from capture to shift cycle.

[0023] The only difference between these two techniques is that the former uses clock edges to trigger the next operation, the latter uses signal levels to enable the next operation. In practice, a mixed approach can be employed. Since a daisy-chain or token-ring approach is used, the multiple-capture DFT system allows testing of any frequency domain at a reduced clock speed when this particular frequency domain cannot operate at-speed. This is very common in testing high-speed integrated circuits, such as microprocessors and networking chips, where different clock speeds of chips are sold at different prices. In addition, due to its ease of control, this approach further allows at-speed scan-test simply using internally reconfigured capture clocks. Thus, a low-cost tester (ATE) can be used for at-speed scan-test, in addition to at-speed self-test.

[0024] The multiple-capture DFT system in the present invention further comprises applying an ordered sequence of capture clocks and operating each capture clock at its selected clock speed in the capture operation (cycle). The ordered sequence of capture clocks is applied to the circuit under test one-by-one using the daisy-chain clock-triggering or token-ring clock-enabling technique. The order of these capture clocks is further programmable, when it's required to increase the circuit's fault coverage. Each capture clock can be also disabled or chosen to facilitate fault diagnosis. In addition, when two clock domains do not interact with each other, they can be tested simultaneously to shorten the capture cycle time.

[0025] Each capture clock of the present invention further comprises one or more clock pulses. The number of clock pulses is further programmable. When self-test is employed, the multiple-capture DFT system is usually placed inside the integrated circuit and, thus, all capture clocks are generated internally. When scan-test is employed, the multiple-capture DFT system is usually resided in an ATE and, thus, all capture clocks are controlled externally. However, for at-speed scan-test, it's often required to capture output responses using its respective operating frequency within each clock domain. The present invention further comprises any method or apparatus for allowing use of internally-generated or externally-controlled capture clocks for at-speed scan-test or self-test.

[0026] After the capture operation is completed, all output responses captured at all scan cells are compacted internally to signatures or shifted out to the multiple-capture DFT system for direct comparison. The compact or compare operation occurs concurrently with the shift operation, and the process of shift, capture, and compact/compare operations shall continue until a predetermined limiting criteria, such as completion of all self-test or scan-test cycles, is reached. Finally, the multiple-capture DFT system will compare the signatures against expected signatures when the compact operation is employed during self-test or scan-test. Such comparison can be done either in the integrated circuit with a built-in comparator or in an ATE by shifting the final signatures out for analysis.

[0027] In the present invention, both self-test and scan-test techniques are employed to detect or locate stuck-at and delay faults. The stuck-at faults further comprise other stuck-type faults, such as open and bridging faults. The delay faults further comprise other non-stuck-type delay faults, such as transition (gate-delay), multiple-cycle delay, and path-delay faults. In addition, each scan cell can be a multiplexed D flip-flop or a level sensitive latch, and the integrated circuit or circuit assembly under test can be a full-scan or partial-scan design.

[0028] In general, it is only required to apply one clock pulse and two consecutive clock pulses to test stuck-at faults and delay faults within one clock domain, respectively. Multiple-cycle paths present within one clock domain and between clock domains, however, require waiting for a number of clock cycles for capturing. To test multiple-cycle paths within clock domains, the present invention further comprise applying only one clock pulse to test these multiple-cycle paths within each clock domain by reducing the frequency of that domain's capture clock speed to the level where only paths of equal cycle latency (cycle delays) are captured at its intended rated clock speed one at a time. To test multiple-cycle paths between two clock domains, the present invention further comprise adjusting the relative clock delay along the paths to the level where the crossing-boundary multiple-cycle paths are captured at its intended rated clock speed.

[0029] To summarize, the present invention centers on using one global scan enable (GSE) signal for driving all scan enable (SE) signals at a reduced clock speed and applying an ordered sequence of capture clocks for capturing output responses in both self-test and scan-test modes. The present invention assumes that the integrated circuit or circuit assembly must contain two or more clock domains

each controlled by one capture clock. During self-test, each capture clock shall contain one or more clock pulses, and during scan-test, one of the capture clocks must contain two or more clock pulses.

[0030] Due to its ease of control on the scan enable and capture clock signals, the multiple-capture DFT system of the present invention can now be easily realized by an apparatus and synthesized using computer-aided design (CAD) tools. The present invention further comprises such a CAD system for synthesizing the apparatus and verifying its correctness using combinational fault simulation and combinational ATPG in self-test or scan-test mode.

BRIEF DESCRIPTION OF THE DRAWINGS

[0031] The above and other objects, advantages and features of the invention will become more apparent when considered with the following specification and accompanying drawings wherein:

[0032] **FIG. 1** shows an example full-scan or partial-scan design with 4 clock domains and 4 system clocks, where a multiple-capture DFT system in accordance with the present invention is used to detect or locate stuck-at faults at a reduced clock speed in self-test or scan-test mode.

[0033] **FIG. 2** shows a multiple-capture DFT system with multiple PRPG-MISR pairs, in accordance with the present invention, which is used at a reduced clock speed in self-test mode to detect or locate stuck-at faults in the design given in **FIG. 1**.

[0034] **FIG. 3** shows a timing diagram of the full-scan design given in **FIG. 1**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate stuck-at faults within each clock domain and stuck-at faults crossing clock domains in self-test mode. The chain of control events is also shown.

[0035] **FIG. 4** shows a timing diagram of the full-scan design given in **FIG. 1**, in accordance with the present invention, where a shortened yet ordered sequence of capture clocks is used to detect or locate stuck-at faults within each clock domain and stuck-at faults crossing clock domains in self-test mode.

[0036] **FIG. 5** shows a timing diagram of the full-scan design given in **FIG. 1**, in accordance with the present invention, where an expanded yet ordered sequence of capture clocks is used to detect or locate other stuck-type faults within each clock domain and other stuck-type faults crossing clock domains in self-test or scan-test mode.

[0037] **FIG. 6** shows a timing diagram of the partial-scan design given in **FIG. 1**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate stuck-at faults within each clock domain and stuck-at faults crossing clock domains in self-test or scan-test mode.

[0038] **FIG. 7** shows an example full-scan or partial-scan design with 4 clock domains and 4 system clocks, where a multiple-capture DFT system in accordance with the present invention is used to detect or locate stuck-at, delay, and multiple-cycle delay faults at its desired clock speed in self-test or scan-test mode.

[0039] **FIG. 8** shows a multiple-capture DFT system with multiple PRPG-MISR pairs, in accordance with the present invention, which is used at its desired clock speed in self-test or scan-test mode to detect or locate stuck-at, delay, and multiple-cycle delay faults in the design given in **FIG. 7**.

[0040] **FIG. 9** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate stuck-at faults within each clock domain and stuck-at faults crossing clock domains in self-test mode.

[0041] **FIG. 10** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate delay faults within each clock domain and stuck-at faults crossing clock domains in self-test or scan-test mode. The chain of control events is also shown.

[0042] **FIG. 11** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where a shortened yet ordered sequence of capture clocks is used to detect or locate delay faults within each clock domain and stuck-at faults crossing clock domains in self-test or scan-test mode.

[0043] **FIG. 12** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate stuck-at faults within each clock domain and delay faults crossing clock domains in self-test or scan-test mode.

[0044] **FIG. 13** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate delay faults within each clock domain and delay faults crossing clock domains in self-test or scan-test mode.

[0045] **FIG. 14** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where a reordered sequence of capture clocks is used to detect or locate delay faults within each clock domain and stuck-at faults crossing clock domains in self-test or scan-test mode.

[0046] **FIG. 15** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where an expanded yet ordered sequence of capture clocks is used to detect or locate additional delay faults within each clock domain and additional stuck-at faults crossing clock domains in self-test or scan-test mode.

[0047] **FIG. 16** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate 2-cycle delay faults within each clock domain and stuck-at faults crossing clock domains in self-test or scan-test mode.

[0048] **FIG. 17** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate 2-cycle delay faults within each clock domain and 2-cycle delay faults crossing clock domains in self-test or scan-test mode.

[0049] **FIG. 18** shows a timing diagram of the partial-scan design given in **FIG. 7**, in accordance with the present

invention, where an ordered sequence of capture clocks is used to detect or locate stuck-at faults within each clock domain and stuck-at faults crossing clock domains in self-test or scan-test mode.

[0050] **FIG. 19** shows a timing diagram of the partial-scan design given in **FIG. 7**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate delay faults within each clock domain and stuck-at faults crossing clock domains in self-test or scan-test mode.

[0051] **FIG. 20** shows a timing diagram of the partial-scan design given in **FIG. 7**, in accordance with the present invention, where an ordered sequence of capture clocks is used to detect or locate 2-cycle delay faults within each clock domain and stuck-at faults crossing clock domains in self-test or scan-test mode.

[0052] **FIG. 21** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where the capture clock **CK2** during the capture cycle is chosen to diagnose faults captured by **CK2** in self-test or scan-test mode.

[0053] **FIG. 22** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where the capture clocks **CK1** and **CK3** during the capture cycle are chosen to diagnose faults captured by **CK1** and **CK3** in self-test or scan-test mode.

[0054] **FIG. 23** shows a timing diagram of the full-scan design given in **FIG. 1**, in accordance with the present invention, where all capture clocks during the shift cycle are skewed to reduce power consumption.

[0055] **FIG. 24** shows a multiple-capture CAD system in accordance with the present invention, where a CAD system is used to implement the multiple-capture DFT technique on a full-scan or partial-scan design in self-test mode.

[0056] **FIG. 25** shows a multiple-capture CAD system in accordance with the present invention, where a CAD system is used to implement the multiple-capture DFT technique on a full-scan or partial-scan design in scan-test mode.

DETAILED DESCRIPTION OF THE DRAWINGS

[0057] The following description is of presently contemplated as the best mode of carrying out the present invention. This description is not to be taken in a limiting sense but is made merely for the purpose of describing the principles of the invention. The scope of the invention should be determined by referring to the appended claims.

[0058] **FIG. 1** shows an example full-scan or partial-scan design with a multiple-capture DFT system, of one embodiment of the invention. The design **133** contains 4 clock domains, **CD1102** to **CD4105**, and 4 system clocks, **CK1111** to **CK4120**. Each system clock controls one clock domain. **CD1102** and **CD2103** talk to each other via a crossing clock-domain logic block **CCD1106**; **CD2103** and **CD3104** talk to each other via a crossing clock-domain logic block **CCD2107**; and **CD3104** and **CD4105** talk to each other via a crossing clock-domain logic block **CCD3108**.

[0059] The 4 clock domains, **CD1102** to **CD4105**, are originally designed to run at 150 MHz, 100 MHz, 100 MHz, and 66 MHz, respectively. However, in this example, since a DFT (self-test or scan-test) technique is only employed to detect or locate stuck-at faults in the design **133**, all system clocks, **CK1111** to **CK4120**, are reconfigured to operate at 10 MHz. The reconfigured system clocks are called capture clocks.

[0060] During self-test or scan-test, the multiple-capture DFT system **101** will take over the control of all stimuli, **109**, **112**, **115**, and **118**, all system clocks, **CK1111** to **CK4120**, and all output responses, **110**, **113**, **116**, and **119**.

[0061] During the shift operation, the multiple-capture DFT system **101** first generates and shifts pseudorandom or predetermined stimuli through **109**, **112**, **115**, and **118** to all scan cells SC in all scan chains SCN within the 4 clock domains, **CD1102** to **CD4105**, simultaneously. The multiple-capture DFT system **101** shall wait until all stimuli, **109**, **112**, **115**, and **118**, have been shifted into all scan cells SC. It should be noted that, during the shift operation, the capture clock can be operated either at its rated clock speed (at-speed) or at a desired clock speed.

[0062] After the shift operation is completed, an ordered sequence of capture clocks is applied to all clock domains, **CD1102** to **CD4105**. During the capture operation, each capture clock can operate at its rated clock speed (at-speed) or at a reduced speed (slow-speed), and can be generated internally or controlled externally. In this example, all system clocks, **CK1111** to **CK4120**, are reconfigured to operate at a reduced frequency of 10 MHz.

[0063] After the capture operation is completed, the output responses captured at all scan cells SC are shifted out through responses **110**, **113**, **116**, and **119** to the multiple-capture DFT system **101** for compaction during the compact operation or direct comparison during the compare operation.

[0064] Based on **FIG. 1**, the timing diagrams given in FIGS. **3** to **6** are used to illustrate that, by properly ordering the sequence of capture clocks and by adjusting relative inter-clock delays, stuck-at faults within each clock domain and crossing clock domains can be detected or located in self-test or scan-test mode. Please note that different ways of ordering the sequence of capture clocks and adjusting relative inter-clock delays will result in different faults to be detected or located.

[0065] **FIG. 2** shows a multiple-capture DFT system with three PRPG-MISR pairs, of one embodiment of the invention, used to detect or locate stuck-at faults in the design **133** given in **FIG. 1** in self-test mode.

[0066] Pseudorandom pattern generators (PRPGs), **211** to **213**, are used to generate pseudorandom patterns. Phase shifters, **214** to **216**, are used to break the dependency between different outputs of the PRPGS. The bit streams coming from the phase shifters become test stimuli, **109**, **112**, **115**, and **118**.

[0067] Space compactors, **217** to **219**, are used to reduce the number of bit streams in test responses, **110**, **113**, **116**, and **119**. Space compactors are optional and are only used when the overhead of a MISR becomes a concern. The outputs of the space compactors are then compressed by multiple input signature registers (MISRs), **220** to **222**. The contents of MISRs after all test stimuli are applied become signatures, **236** to **238**. The signatures are then be compared

by comparators, **223** to **225**, with corresponding expected values. The error indicator **226** is used to combine the individual pass/fail signals, **242** to **244**, a global pass/fail signal **245**. Alternatively, the signatures in MISRs **220** to **222** can be shifted to the outside of the design for comparison through a single scan chain composed of elements **223**, **239**, **224**, **240**, **225**, and **241**.

[0068] The central self-test controller **202** controls the whole test process by manipulating individual scan enable signals, **204** to **207**, and by reconfiguring capture clocks, CK1111 to CK4120. Especially, the scan enable signals, **204** to **207**, can be controlled by one global scan enable signal GSE **201**, which can be a slow signal in that it does not have to settle down in half of the cycle of any clock applied to any clock domain. Some additional control signals **203** are needed to conduct other control tasks.

[0069] The clock domains **103** and **104**, which are operated at the same frequency, share the same pair of PRPG **212** and MISR **221**. It should be noted that the skew between the clocks CK2114 and CK3117 should be properly managed to prevent any timing violations during the shift operation and any races during the capture operation.

[0070] All storage elements in PRPGs, **211** to **213**, and MISRs, **220** to **222**, can be connected into a scan chain from which predetermined patterns can be shifted in for reseeding and computed signatures can be shifted out for analysis. This configuration helps in increasing fault coverage and in facilitating fault diagnosis.

[0071] **FIG. 3** shows a timing diagram of a full-scan design given in **FIG. 1**, of one embodiment of the invention for detecting or locating stuck-at faults within each clock domain and stuck-at faults crossing clock domains with an ordered sequence of capture clocks in self-test mode. The timing diagram **300** shows the sequence of waveforms of the 4 capture clocks, CK1111 to CK4120, operating at the same frequency.

[0072] During each shift cycle **310**, a series of pulses of 10 MHz are applied through capture clocks, CK1111 to CK4120, to shift stimuli to all scan cells within all clock domains, CD1102 to CD4105.

[0073] During each capture cycle **311**, 4 sets of capture clock pulses are applied in the following order: First, one capture pulse is applied to CK1111 to detect or locate stuck-at faults within the clock domain CD1102. Second, one capture pulse is applied to CK2114 to detect or locate stuck-at faults within the clock domain CD2103. Third, one capture pulse is applied to CK3117 to detect or locate stuck-at faults within the clock domain CD3104. Fourth, one capture pulse is applied to CK4120 to detect or locate stuck-at faults within the clock domain CD4105.

[0074] In addition, the stuck-at faults which can be reached from lines **121**, **125**, and **129** in the crossing clock-domain logic blocks CCD1106 to CCD3108, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay **307** between the rising edge of the capture pulse of CK1111 and the rising edge of the capture pulse of CK2114 must be adjusted so that no races or timing violations would occur while the output responses **123** are captured through the crossing clock-domain logic block CCD1106.

[0075] The same principle applies to the relative clock delay **308** between CK2114 and CK3117, and the relative clock delay **309** between CK3117 and CK4120 for capturing output responses, **127** and **131**, through CCD2107 and CCD3108, respectively.

[0076] It should be noticed that, generally, during each shift cycle, any capture clock is allowed to operate at its desired or a reduced clock speed. In addition, it is not necessary that all capture clocks must operate at the same clock speed. Furthermore, to reduce peak power consumption during the shift cycle, all capture clocks can be skewed so that at any given time only scan cells within one clock domain can change states. One global scan enable signal GSE **201**, operated at a reduced clock speed, can also be used, when requested, to switch the test operation from the shift cycle to the capture cycle, and vice versa.

[0077] The daisy-chain clock-triggering technique is used to generate and order the sequence of capture clocks one after the other in the following way: The rising edge of the last pulse in the shift cycle triggers the event **301** of applying **0** to the global scan enable GSE **201**, switching the test operation from the shift cycle to the capture cycle. The falling edge of GSE **201** triggers the event **302** of applying one capture pulse to CK1111. Similarly, the rising edge of the capture pulse of CK1111 triggers the event **303** of applying one capture pulse to CK2114, the rising edge of the capture pulse of CK2114 triggers the event **304** of applying one capture pulse to CK3117, and the rising edge of the capture pulse of CK3117 triggers the event **305** of applying one capture pulse to CK4120. Finally, the rising edge of the capture pulse of CK4120 triggers the event **306** of applying **1** to the global scan enable GSE **201**, switching the test operation from the capture cycle to the shift cycle. This daisy-chain clock-triggering technique is also used to order the sequence of capture clocks in FIGS. **4** to **6**.

[0078] **FIG. 4** shows a timing diagram of a full-scan design given in **FIG. 1**, of one embodiment of the invention for detecting or locating stuck-at faults within each clock domain and stuck-at faults crossing clock domains with a shortened yet ordered sequence of capture clocks in self-test mode. The timing diagram **400** shows the sequence of waveforms of the 4 capture clocks, CK1111 to CK4120, operating at the same frequency.

[0079] During each shift cycle **402**, a series of clock pulses of 10 MHz are applied through capture clocks, CK1111 to CK4120, to shift stimuli to all scan cells within all clock domains, CD1102 to CD4105.

[0080] During each capture cycle **403**, two sets of capture clock pulses are applied in the following order: First, one capture pulse is applied to CK1111 and CK3117 simultaneously to detect or locate stuck-at faults within the clock domain CD1102 and CD3104, respectively. Second, one capture pulse is applied to CK2114 and CK4120 simultaneously to detect or locate stuck-at faults within the clock domain CD2103 and CD4105, respectively.

[0081] In addition, the stuck-at faults which can be reached from lines **121**, **128**, and **129** in the crossing clock-domain logic blocks CCD1106 to CCD3108, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay **401** between the rising edge of the capture pulse for CK1111 and

CK3117 and the rising edge of the capture pulse for CK2114 and CK4120, must be adjusted so that no races or timing violations would occur while the output responses, 123, 126, and 131, are captured through the crossing clock-domain logic blocks CCD1106 to CCD3108.

[0082] FIG. 5 shows a timing diagram of a full-scan design in FIG. 1 of one embodiment of the invention for detecting or locating other stuck-type faults within each clock domain and other stuck-type faults crossing clock domains with an expanded yet ordered sequence of capture clocks in self-test or scan-test mode. The timing diagram 500 shows the sequence of waveforms of the 4 capture clocks, CK1111 to CK4120, operating at the same frequency.

[0083] During each shift cycle 503, a series of clock pulses of 10 MHz are applied through capture clocks, CK1111 to CK4120, to shift stimuli to all scan cells within all clock domains, CD1102 to CD4105.

[0084] During each capture cycle 504, two sets of capture clock pulses are applied in the following order: First, two capture pulses are applied to CK1111 and CK3117, simultaneously. Second, one capture pulse is applied to CK2114 and CK4120, simultaneously. Stuck-at faults in all crossing clock-domain combinations, from 121 to 123, from 124 to 122, from 125 to 127, from 128 to 126, from 129 to 131, from 132 to 130, can be detected or located if the following condition is satisfied: The relative clock delay 501 between the rising edge of the first capture pulse of CK1111 and CK3117 and the rising edge of the capture pulse of CK2114 and CK4120 must be adjusted so that no races or timing violations would occur while the output responses 123, 126, and 131 are captured through the crossing clock-domain logic block CCD1106 to CCD3108, respectively. The relative clock delay 502 between the rising edge of the capture pulse of CK2114 and CK4120 and the second capture pulse of CK1111 and CK3117 must be adjusted so that no races or timing violations would occur while the output responses 122, 127, and 130 are captured through the crossing clock-domain logic block CCD1106 to CCD3108, respectively.

[0085] FIG. 6 shows a timing diagram of a feed-forward partial-scan design given in FIG. 1, of one embodiment of the invention for detecting or locating stuck-at faults within each clock domain and stuck-at faults crossing clock domains with a shortened yet ordered sequence of capture clocks in self-test or scan-test mode. It is assumed that the clock domains CD1102 to CD4105 contain a number of un-scanned storage cells that form a sequential depth of no more than 2. The timing diagram 600 shows the sequence of waveforms of the 4 capture clocks, CK1111 to CK4120, operating at the same frequency.

[0086] During each shift cycle 606, a series of clock pulses of 10 MHz are applied through capture clocks, CK1111 to CK4120, to shift stimuli to all scan cells within all clock domains, CD1102 to CD4105.

[0087] During each capture cycle 607, two sets of capture clock pulses are applied in the following order: First, three pulses of 10 MHz, two being functional pulses and one being a capture pulse, are applied to CK1111 and CK3117 simultaneously to detect or locate stuck-at faults within the clock domain CD1102 and CD3104, respectively. Second, three pulses of 10 MHz, two being functional pulses and one being a capture pulse, are applied to CK2114 and CK4120 simultaneously to detect or locate stuck-at faults within the clock domain CD2103 and CD4105, respectively.

[0088] In addition, the stuck-at faults which can be reached from lines 121, 128, and 129 in the crossing clock-domain logic blocks CCD1106 to CCD3108, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay 603 between the rising edge of the capture pulse for CK1111 and CK3117 and the rising edge of the capture pulse for CK2114 and CK4120 must be adjusted so that no races or timing violations would occur while the output responses, 123, 126, and 131, are captured through the crossing clock-domain logic blocks CCD1106 to CCD3108.

[0089] FIG. 7 shows an example full-scan or partial-scan design with a multiple-capture DFT system, of one embodiment of the invention. The design 733 is the same as the design 133 given in FIG. 1. Same as in FIG. 1, the 4 clock domains, CD1702 to CD4705, are originally designed to run at 150 MHz, 100 MHz, 100 MHz, and 66 MHz, respectively. The only difference from FIG. 1 is that these clock frequencies will be used directly without alternation in order to implement at-speed self-test or scan-test for stuck-at, delay, and multiple-cycle delay faults within each clock domain and crossing clock domains.

[0090] Based on FIG. 7, the timing diagrams given in FIGS. 9 to 20 are used to illustrate that, by properly ordering the sequence of capture pulses and by adjusting relative inter-clock delays, the at-speed detection or location of stuck-at, delay, and multiple-cycle delay faults within each clock domain and crossing clock domains can be achieved in self-test or scan-test mode. Please note that different ways of ordering the sequence of capture pulses and adjusting relative inter-clock delays will result in different faults to be detected or located.

[0091] FIG. 8 shows a multiple-capture DFT system with three PRPG-MISR pairs, of one embodiment of the invention, used in self-test or scan-test mode to detect or locate stuck-at, delay, and multiple-cycle delay faults in the design given in FIG. 7. The composition and operation of the multiple-capture DFT system is basically the same as the one given in FIG. 2. There are two major differences: One is that, in this example, the original clock frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are used directly without alternation in order to implement at-speed self-test or scan-test. The other is that more care needs to be taken in the physical design of scan chains, etc., in this example.

[0092] The clock domains 703 and 704, which are operated at the same frequency, share the same pair of PRPG 812 and MISR 821. It should be noted that the skew between the clocks CK2714 and CK3717 should be properly managed to prevent any timing violations during the shift operation and any races during the capture operation.

[0093] All storage elements in PRPGs, 811 to 813, and MISRs, 820 to 822, can be connected into a scan chain from which predetermined patterns can be shifted in for reseeding and computed signatures can be shifted out for analysis. This configuration helps in increasing fault coverage and in facilitating fault diagnosis.

[0094] FIG. 9 shows a timing diagram of a full-scan design given in FIG. 7, of one embodiment of the invention

for detecting or locating stuck-at faults within each clock domain and stuck-at faults crossing clock domains with an ordered sequence of capture clocks in self-test mode. The timing diagram **900** shows the sequence of waveforms of the 4 capture clocks, CK**1711** to CK**4720**, operating at different frequencies. This timing diagram is basically the same as the one given in **FIG. 3** except the capture clocks, CK**1711** to CK**4720**, run at 150 MHz, 100 MHz, 100 MHz, and 66 MHz, respectively, in both shift and capture cycles, instead of 10 MHz as in **FIG. 3**.

[**0095**] **FIG. 10** shows a timing diagram of a full-scan design given in **FIG. 7**, of one embodiment of the invention for detecting or locating delay faults within each clock domain and stuck-at faults crossing clock domains with an ordered sequence of capture clocks in self-test or scan-test mode. The timing diagram **1000** shows the sequence of waveforms of the 4 capture clocks, CK**1711** to CK**4720**, operating at different frequencies.

[**0096**] During each shift cycle **1014**, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK**1711** to CK**4720**, to shift stimuli to all scan cells within all clock domains, CD**1702** to CD**4705**.

[**0097**] During each capture cycle **1015**, 4 sets of capture clock pulses are applied in the following order: First, two capture pulses of 150 MHz are applied to CK**1711** to detect or locate delay faults within the clock domain CD**1702**. Second, two capture pulses of 100 MHz are applied to CK**2714** to detect or locate delay faults within the clock domain CD**2703**. Third, two capture pulses of 100 MHz are applied to CK**3717** to detect or locate delay faults within the clock domain CD**3704**. Fourth, two capture pulses of 66 MHz are applied to CK**4720** to detect or locate delay faults within the clock domain CD**4705**.

[**0098**] In addition, the stuck-at faults which can be reached from lines **721**, **725**, and **729** in the crossing clock-domain logic blocks CCD**1706** to CCD**3708**, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay **1008** between the rising edge of the second capture pulse of CK**1711** and the rising edge of the first capture pulse of CK**2714** must be adjusted so that no races or timing violations would occur while the output responses **723** are captured through the crossing clock-domain logic block CCD**1706**.

[**0099**] The same principle applies to the relative clock delay **1010** between CK**2714** and CK**3717**, and the relative clock delay **1012** between CK**3717** and CK**4720** for capturing the output responses, **727** and **731**, through CCD**2707** and CCD**3708**, respectively.

[**0100**] The daisy-chain clock-triggering technique is used to generate and order the sequence of capture clocks one after the other in the following way: The rising edge of the last pulse in the shift cycle triggers the event **1001** of applying 0 to the global scan enable GSE **801**, switching the test operation from the shift cycle to the capture cycle. The falling edge of GSE **801** triggers the event **1002** of applying two capture pulses to CK**1711**. Similarly, the rising edge of the second capture pulse of CK**1711** triggers the event **1003** of applying two capture pulses to CK**2714**, the rising edge of the second capture pulse of CK**2714** triggers the event **1004** of applying two capture pulses to CK**3717**, and the rising edge of the second capture pulse of CK**3717** triggers the event **1005** of applying two capture pulses to CK**4720**. Finally, the rising edge of the second capture pulse of CK**4720** triggers the event **1006** of applying 1 to the global scan enable GSE **801**, switching the test operation from the capture cycle to the shift cycle. This daisy-chain clock-triggering technique is also used to order the sequence of capture clocks in **FIG. 9** and FIGS. **11** to **20**.

[**0101**] **FIG. 11** shows a timing diagram of a full-scan design given in **FIG. 7**, of one embodiment of the invention for detecting or locating delay faults within each clock domain and stuck-at faults crossing clock domains with a shortened yet ordered sequence of capture clocks in self-test or scan-test mode. The timing diagram **1100** shows the sequence of waveforms of the 4 capture clocks, CK**1711** to CK**4720**, operating at different frequencies.

[**0102**] During each shift cycle **1108**, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK**1711** to CK**4720**, to shift stimuli to all scan cells within all clock domains, CD**1702** to CD**4705**.

[**0103**] During each capture cycle **1109**, 4 sets of capture clock pulses are applied in the following order: First, two capture pulses of frequency 150 MHz are applied to CK**1711** and two clock pulses of frequency 100 MHz are applied to CK**3717**, simultaneously, to detect or locate delay faults within the clock domain CD**1702** and CD**3704**, respectively. Second, two capture pulses of frequency 100 MHz are applied to CK**2714** and two capture pulses of frequency 66 MHz are applied to CK**4720**, simultaneously, to detect or locate delay faults within the clock domain CD**2703** and CD**4705**, respectively.

[**0104**] In addition, the stuck-at faults which can be reached from lines **721**, **728**, and **729** in the crossing clock-domain logic blocks CCD**1706** to CCD**3708**, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay **1102** between the rising edge of the second capture pulse of CK**1711** and the rising edge of the first capture pulse of CK**2714** must be adjusted so that no races or timing violations would occur while the output responses **723** are captured through the crossing clock-domain logic block CCD**1706**.

[**0105**] The same principle applies to the relative clock delay **1104** between CK**3717** and CK**2714**, and the relative clock delay **1106** between CK**3717** and CK**4720** for capturing the output responses, **726** and **731**, through CCD**2707** and CCD**3708**, respectively.

[**0106**] **FIG. 12** shows a timing diagram of a full-scan design given in **FIG. 7**, of one embodiment of the invention for detecting or locating stuck-at faults within each clock domain and delay faults crossing clock domains with an ordered sequence of capture clocks in self-test or scan-test mode. The timing diagram **1200** shows the sequence of waveforms of the 4 capture clocks, CK**1711** to CK**4720**, operating at different frequencies.

[**0107**] During each shift cycle **1204**, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks,

CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0108] During each capture cycle 1205, 4 sets of capture clock pulses are applied in the following order: First, one capture pulse of 150 MHz is applied to CK1711 to detect or locate stuck-at faults within the clock domain CD1702. Second, one capture pulse of 100 MHz is applied to CK2714 to detect or locate stuck-at faults within the clock domain CD2703. Third, one capture pulse of 100 MHz is applied to CK3717 to detect or locate stuck-at faults within the clock domain CD3704. Fourth, one capture pulse of 66 MHz is applied to CK4720 to detect or locate stuck-at faults within the clock domain CD4705.

[0109] In addition, the delay faults which can be reached from lines 721, 725, and 729 in the crossing clock-domain logic blocks CCD1706 to CCD3708, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delays 1201 between the rising edge of the capture pulse of CK1711 and the rising edge of the capture pulse of CK2714 must be adjusted to meet the at-speed timing requirements for paths from 721 to 723. Similarly, the relative clock delay 1202 between CK2714 and CK3717, and the relative clock delay 1203 between CK3717 and CK4720, must be adjusted to meet the at-speed timing requirements for paths from 725 to 727, and paths from 729 to 731, respectively.

[0110] FIG. 13 shows a timing diagram of a full-scan design given in FIG. 7, of one embodiment of the invention for detecting or locating delay faults within each clock domain and delay faults crossing clock domains with an ordered sequence of capture clocks in self-test or scan-test mode. The timing diagram 1300 shows the sequence of waveforms of the 4 capture clocks, CK1711 to CK4720, operating at different frequencies.

[0111] During each shift cycle 1308, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0112] During each capture cycle 1309, 4 sets of capture clock pulses are applied in the following order: First, two capture pulses of 150 MHz are applied to CK1711 to detect or locate delay faults within the clock domain CD1702. Second, two capture pulses of 100 MHz are applied to CK2714 to detect or locate delay faults within the clock domain CD2703. Third, two capture pulses of 100 MHz are applied to CK3717 to detect or locate delay faults within the clock domain CD3704. Fourth, two capture pulses of 66 MHz are applied to CK4720 to detect or locate delay faults within the clock domain CD4705.

[0113] In addition, the delay faults which can be reached from lines 721, 725, and 729 in the crossing clock-domain logic blocks CCD1706 to CCD3708, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay 1302 between the rising edge of the second capture pulse of CK1711 and the rising edge of the first capture pulse of CK2714 must be adjusted to meet the at-speed timing requirements for paths from 721 to 723. Similarly, the relative clock delay 1304 between CK2714 and CK3717, and the relative clock delay 1306 between CK3717 and CK4720, must be adjusted to meet the at-speed timing requirements for paths from 725 to 727, and paths from 729 and 731, respectively.

[0114] FIG. 14 shows a timing diagram of a full-scan design given in FIG. 7, of one embodiment of the invention for detecting or locating delay faults within each clock domain and stuck-at faults crossing clock domains with a reordered sequence of capture clocks in self-test or scan-test mode. The timing diagram 1400 shows the sequence of waveforms of the 4 capture clocks, CK1711 to CK4720, operating at different frequencies.

[0115] During each shift cycle 1408, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0116] During each capture cycle 1409, 4 sets of capture clock pulses are applied in the following order: First, two capture pulses of 66 MHz are applied to CK4720 to detect or locate delay faults within the clock domain CD4705. Second, two capture pulses of 100 MHz are applied to CK3717 to detect or locate delay faults within the clock domain CD3704. Third, two capture pulses of 100 MHz are applied to CK2714 to detect or locate delay faults within the clock domain CD2703. Fourth, two capture pulses of 150 MHz are applied to CK1711 to detect or locate delay faults within the clock domain CD1702.

[0117] In addition, the stuck-at faults which can be reached from lines 724, 728, and 732 in the crossing clock-domain logic blocks CCD1706 to CCD3708, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay 1402 between the rising edge of the second capture pulse of CK4720 and the rising edge of the first capture pulse of CK3717 must be adjusted so that no races or timing violations would occur while the output responses 730 are captured through the crossing clock-domain logic block CCD3708.

[0118] The same principle applies to the relative clock delay 1404 between CK3717 and CK2714, and the relative clock delay 1406 between CK2714 and CK1711 for capturing output responses, 726 and 722, through CCD2707 and CCD1706, respectively.

[0119] FIG. 15 shows a timing diagram of a full-scan design given in FIG. 7, of one embodiment of the invention for detecting or locating additional delay faults within each clock domain and additional stuck-at faults crossing clock domains with an expanded yet ordered sequence of capture clocks in self-test or scan-test mode. The timing diagram 1500 shows the sequence of waveforms of the 4 capture clocks, CK1711 to CK4720, operating at different frequencies.

[0120] During each shift cycle 1514, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0121] During each capture cycle 1515, seven sets of double-capture pulses are applied in the following order: First, two capture pulses of 150 MHz are applied to CK1711. Second, two capture pulses of 100 MHz are applied to

CK2714. Third, two capture pulses of 100 MHz are applied to CK3717. Fourth, two capture pulses of 66 MHz are applied to CK4720. Fifth, two capture pulses of 100 MHz are applied to CK3717. Sixth, two capture pulses of 100 MHz are applied to CK2714. Seventh, two capture pulses of 150 MHz are applied to CK1711.

[0122] For the capture clock CK1711, the second pulse and the third pulse are used to launch the transition needed for detecting or locating delay faults within the clock domain CD1702. Since the transition is generated by two close-to-functional patterns, the risk of activating a false path is lower. In addition, additional delay faults within the clock domain CD1702 can be detected or located by the transition. The same results also apply to the clock domains CD2703 and CD3704.

[0123] In addition, the stuck-at faults which can be reached from lines 724, 728, and 732 in the crossing clock-domain logic blocks CCD1706 to CCD3708, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay 1508 between the rising edge of the second capture pulse of CK4720 and the rising edge of the first capture pulse of CK3717 must be adjusted so that no races or timing violations would occur while the output responses 730 are captured through the crossing clock-domain logic block CCD3708.

[0124] The same principle applies to the relative clock delay 1510 between CK3717 and CK2714, and the relative clock delay 1512 between CK2714 and CK1711 for capturing output responses, 726 and 722, through CCD2707 and CCD1706, respectively.

[0125] FIG. 16 shows a timing diagram of a full-scan design given in FIG. 7, of one embodiment of the invention for detecting or locating 2-cycle delay faults within each clock domain and stuck-at faults crossing clock domains with an ordered sequence of capture clocks in self-test or scan-test mode. It is assumed that some paths in the clock domains, CD1702 to CD4705, need two cycles for signals to pass through. The timing diagram 1600 shows the sequence of waveforms of the 4 capture clocks, CK1711 to CK4720, operating at different frequencies.

[0126] During each shift cycle 1608, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0127] During each capture cycle 1609, 4 sets of capture clock pulses are applied in the following order: First, two capture pulses of 75 MHz (half of 150 MHz) are applied to CK1711 to detect or locate 2-cycle delay faults within the clock domain CD1702. Second, two capture pulses of 50 MHz (half of 100 MHz) are applied to CK2714 to detect or locate 2-cycle delay faults within the clock domain CD2703. Third, two capture pulses of 50 MHz (half of 100 MHz) are applied to CK3717 to detect or locate 2-cycle delay faults within the clock domain CD3704. Fourth, two capture pulses of 33 MHz (half of 66 MHz) are applied to CK4720 to detect or locate 2-cycle delay faults within the clock domain CD4705.

[0128] In addition, the stuck-at faults which can be reached from lines 721, 725, and 729 in the crossing clock-domain logic blocks CCD1706 to CCD3708, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay 1602 between the rising edge of the second capture pulse of CK1711 and the rising edge of the first capture pulse of CK2714 must be adjusted so that no races or timing violations would occur while the output responses 723 are captured through the crossing clock-domain logic block CCD1706.

[0129] The same principle applies to the relative clock delay 1604 between CK2714 and CK3717, and the relative clock delay 1606 between CK3717 and CK4720 for capturing output responses, 727 and 731, through CCD2707 and CCD3708, respectively.

[0130] FIG. 17 shows a timing diagram of a full-scan design given in FIG. 7, of one embodiment of the invention for detecting or locating 2-cycle delay faults within each clock domain and 2-cycle delay faults crossing clock domains with an ordered sequence of capture clocks in self-test or scan-test mode. It is assumed that some paths in the clock domains, CD1702 to CD4705, and the crossing clock-domain logic blocks, CCD1706 to CCD3708, need two cycles for signals to pass through. The timing diagram 1700 shows the sequence of waveforms of the 4 capture clocks, CK1711 to CK4720, operating at different frequencies.

[0131] During each shift cycle 1708, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0132] During each capture cycle 1709, 4 sets of capture clock pulses are applied in the following order: First, two capture pulses of 75 MHz (half of 150 MHz) are applied to CK1711 to detect or locate 2-cycle delay faults within the clock domain CD1702. Second, two capture pulses of 50 MHz (half of 100 MHz) are applied to CK2714 to detect or locate 2-cycle delay faults within the clock domain CD2703. Third, two capture pulses of 50 MHz (half of 100 MHz) are applied to CK3717 to detect or locate 2-cycle delay faults within the clock domain CD3704. Fourth, two capture pulses of 33 MHz (half of 66 MHz) are applied to CK4720 to detect or locate 2-cycle delay faults within the clock domain CD4705.

[0133] In addition, the 2-cycle delay faults which can be reached from lines 721, 725, and 729 in the crossing clock-domain logic blocks CCD1706 to CCD3708, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay 1702 between the rising edge of the second capture pulse of CK1711 and the rising edge of the first capture pulse of CK2714 must be adjusted to meet the 2-cycle timing requirements for paths from 721 to 723. Similarly, the relative clock delay 1704 between CK2714 and CK3717, and the relative clock delay 1706 between CK3717 and CK4720, must be adjusted to meet the 2-cycle timing requirements for paths from 725 to 727, and paths from 729 and 731, respectively.

[0134] FIG. 18 shows a timing diagram of a feed-forward partial-scan design given in FIG. 7, of one embodiment of the invention for detecting or locating stuck-at faults within

each clock domain and stuck-at faults crossing clock domains with an ordered sequence of capture clocks in self-test or scan-test mode. It is assumed that the clock domains CD1702 to CD4705 contain a number of unscanned storage cells that form a sequential depth of no more than 2. The timing diagram **1800** shows the sequence of waveforms of the 4 capture clocks, CK1711 to CK4720, operating at different frequencies.

[0135] During each shift cycle **1812**, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0136] During each capture cycle **1813**, 4 sets of capture clock pulses are applied in the following order: First, three pulses of 150 MHz, two being functional pulses and one being a capture pulse, are applied to CK1711 to detect or locate stuck-at faults within the clock domain CD1702. Second, three pulses of 100 MHz, two being functional pulses and one being a capture pulse, are applied to CK2714 to detect or locate stuck-at faults within the clock domain CD2703. Third, three pulses of 100 MHz, two being functional pulses and one being a capture pulse, are applied to CK3717 to detect or locate stuck-at faults within the clock domain CD3704. Fourth, three pulses of frequency 66 MHz, two being functional pulses and one being a capture pulse, are applied to CK4717 to detect or locate stuck-at faults within the clock domain CD4705.

[0137] In addition, the stuck-at faults which can be reached from lines **721**, **725**, and **729** in the crossing clock-domain logic blocks CCD1706 to CCD3708, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay **1803** between the rising edge of the second capture pulse of CK1711 and the rising edge of the first capture pulse of CK2714 must be adjusted so that no races or timing violations would occur while the output responses **723** are captured through the crossing clock-domain logic block CCD1706.

[0138] The same principle applies to the relative clock delay **1806** between CK2714 and CK3717, and the relative clock delay **1809** between CK3717 and CK4720 for capturing output responses, **727** and **731**, through CCD2707 and CCD3708, respectively.

[0139] **FIG. 19** shows a timing diagram of a feed-forward partial-scan design given in **FIG. 7**, of one embodiment of the invention for detecting or locating delay faults within each clock domain and stuck-at faults crossing clock domains with an ordered sequence of capture clocks in self-test or scan-test mode. It is assumed that the clock domains CD1702 to CD4705 contain a number of unscanned storage cells that form a sequential depth of no more than 2. The timing diagram **1900** shows the sequence of waveforms of the 4 capture clocks, CK1711 to CK4720, operating at different frequencies.

[0140] During each shift cycle **1916**, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0141] During each capture cycle **1917**, 4 sets of capture clock pulses are applied in the following order: First, 4 pulses of 150MHZ, two being functional pulses and two being capture pulses, are applied to CK1711 to detect or locate delay faults within the clock domain CD1702. Second, 4 pulses of 100 MHz, two being functional pulses and two being capture pulses, are applied to CK2714 to detect or locate delay faults within the clock domain CD2703. Third, 4 pulses of 100 MHz, two being functional pulses and two being capture pulses, are applied to CK3717 to detect or locate delay faults within the clock domain CD3704. Fourth, 4 pulses of 66 MHz, two being functional pulses and two being capture pulses, are applied to CK4720 to detect or locate delay faults within the clock domain CD4705.

[0142] In addition, the stuck-at faults which can be reached from lines **721**, **725**, and **729** in the crossing clock-domain logic blocks CCD1706 to CCD3708, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay **1904** between the rising edge of the second capture pulse of CK1711 and the rising edge of the first capture pulse of CK2714 must be adjusted so that no races or timing violations would occur while the output responses **723** are captured through the crossing clock-domain logic block CCD1706.

[0143] The same principle applies to the relative clock delay **1908** between CK2714 and CK3717, and the relative clock delay **1912** between CK3717 and CK4720 for capturing output responses, **727** and **731**, through CCD2707 and CCD3708, respectively.

[0144] **FIG. 20** shows a timing diagram of a feed-forward partial-scan design given in **FIG. 7**, of one embodiment of the invention for detecting or locating 2-cycle delay faults within each clock domain and stuck-at faults crossing clock domains with an ordered sequence of capture clocks in self-test or scan-test mode. It is assumed that the clock domains CD1702 to CD4705 contain a number of unscanned storage cells that form a sequential depth of no more than 2. Also, it is assumed that some paths in the clock domains, CD1702 to CD4705, need two cycles for signals to pass through. The timing diagram **2000** shows the sequence of waveforms of the 4 capture clocks, CK1711 to CK4720, operating at different frequencies.

[0145] During each shift cycle **2016**, a series of clock pulses of different frequencies, 150 MHz, 100 MHz, 100 MHz, and 66 MHz, are applied through capture clocks, CK1711 to CK4720, to shift stimuli to all scan cells within all clock domains, CD1702 to CD4705.

[0146] During each capture cycle **2017**, 4 sets of capture clock pulses are applied in the following order: First, 4 pulses, two being functional pulses of 150 MHz and two being capture pulses of 75 MHz (half of 150 MHz), are applied to CK1711 to detect or locate 2-cycle delay faults within the clock domain CD1702. Second, 4 pulses, two being functional pulses of 100 MHz and two being capture pulses of 50 MHz (half of 100 MHz), are applied to CK2714 to detect or locate 2-cycle delay faults within the clock domain CD2703. Third, 4 pulses, two being functional pulses of 100 MHz and two being capture pulses of 50 MHz (half of 100 MHz), are applied to CK3717 to detect or locate 2-cycle delay faults within the clock domain CD3704. Fourth, 4 pulses, 2 being functional pulses of 66 MHz and 2 being capture pulses of 33 MHz (half of 66 MHz), are applied to CK4720 to detect or locate 2-cycle delay faults within the clock domain CD4705.

[0147] In addition, the stuck-at faults which can be reached from lines **721**, **725**, and **729** in the crossing clock-domain logic blocks CCD**1706** to CCD**3708**, respectively, are also detected or located simultaneously if the following condition is satisfied: The relative clock delay **2004** between the rising edge of the second capture pulse of CK**1711** and the rising edge of the first capture pulse of CK**2714** must be adjusted so that no races or timing violations would occur while the output responses **723** are captured through the crossing clock-domain logic block CCD**1706**.

[0148] The same principle applies to the relative clock delay **2008** between CK**2714** and CK**3717**, and the relative clock delay **2012** between CK**3717** and CK**4720** for capturing output responses, **727** and **731**, through CCD**2707** and CCD**3708**, respectively.

[0149] **FIG. 21** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where the capture clock CK2 during the capture cycle is chosen to diagnose faults captured by CK2 in self-test or scan-test mode.

[0150] Fault diagnosis is the procedure by which a fault is located. In order to achieve this goal, it is often necessary to use an approach where a test pattern detects only portion of faults while guaranteeing no other faults are detected. If the test pattern does produce a response that matches the observed response, it can then be declared that the portion must contain at least one actual fault. Then the same approach to the portion of the faults to further localize the actual faults.

[0151] The timing diagram **2100** shows a way to facilitate this approach. In the capture cycle **2107**, two capture pulses of 100 MHz are only applied to the capture clock CK**2714** while the other three capture clocks are held inactive. As a result, for delay faults, only those in the clock domain CD**2703** are detected. In addition, for stuck-at faults, only those in the crossing clock-domain logic blocks CCD**1706** and CCD**2707** and the clock domain CD**2703** are detected. Obviously, this clock timing helps in fault diagnosis.

[0152] **FIG. 22** shows a timing diagram of the full-scan design given in **FIG. 7**, in accordance with the present invention, where the capture clocks CK1 and CK3 during the capture cycle are chosen to diagnose faults captured by CK1 and CK3 in self-test or scan-test mode.

[0153] The diagram **2200** shows one more timing scheme that can help fault diagnosis as described in the description of **FIG. 21**. In the capture cycle **2208**, two capture pulses of 150 MHz are applied to the capture clock CK**1711** and two capture pulses of 100 MHz are applied to the capture clock CK**3717** while the other two capture clocks are held inactive. As a result, for delay faults, only those in the clock domain CD**1702** and CD**3704** are detected. In addition, for stuck-at faults, only those in the crossing clock-domain logic blocks CCD**1706** to CCD**3708** and the clock domains CD**1702** and CD**3703** are detected. Obviously, this clock timing helps in fault diagnosis.

[0154] **FIG. 23** shows a timing diagram of the full-scan design given in **FIG. 1**, in accordance with the present invention, where all capture clocks during the shift cycle are skewed to reduce power consumption. The timing diagram **2300** only shows the waveforms for the capture clocks CK**1111** to CK**4120** during the shift cycle. For the capture cycle, any capture timing control methods claimed in this patent can be applied.

[0155] During the shift cycle **2305**, clock pulses for the clocks CK**1111** to CK**4120** are skewed by properly setting the delay **2301** between the shift pulses for the clocks CK**1111** and CK**2114**, the delay **2302** between the shift pulses for the clocks CK**2114** and CK**3117**, the delay **2303** between the shift pulses for the clocks CK**3117** and CK**4120**, the delay **2304** between the shift pulses for the clocks CK**4120** and CK**1111**. As a result, both peak power consumption and average power consumption are reduced. In addition, during the capture cycle, the PRPG **212** is driven by clock CK**2114**, the first-arrived capture clock, and the MISR **221** is driven by clock CK**3117**, the last-arrived capture clock, in the shared PRPG-MISR pair **228** in **FIG. 2**. Thus, the ordered capture sequence guarantees the correct capture operation when a shared PRPG-MISR pair is used for a plurality of clock domains in self-test mode.

[0156] **FIG. 24** shows a flow chart of one embodiment of the invention. The multiple-capture self-test computer-aided design (CAD) system **2400** accepts the user-supplied HDL code or netlist **2402** together with the self-test control files **2401** and the chosen foundry library **2403**. The self-test control files **2401** contain all set-up information and scripts required for compilation **2404**, self-test rule check **2406**, self-test rule repair **2507**, and multiple-capture self-test synthesis **2408**. As a result, an equivalent combinational circuit model **2409** is generated. Then, combinational fault simulation **2410** can be performed. Finally, post-processing **2411** is used to produce the final self-test HDL code or netlist **2412** as well as the HDL test benches and ATE test programs **2413**. All reports and errors are saved in the report files **2414**.

[0157] The multiple-capture self-test synthesis **2408** uses a hierarchical approach in which it synthesizes a plurality of PRPG-MISR pairs one at a time for each individual clock domain or combined clock domains, then synthesizes a central self-test controller which includes an error indicator, and finally stitches the central self-test controller together with synthesized PRPG-MISR pairs. Each PRPG-MISR pair is composed of a PRPG, an optional phase shifter, an optional space compactor, a MISR, and a comparator. In addition, during PRPG-MISR synthesis, a number of spare scan cells can be inserted into selected clock domains. As a result, the central self-test controller can remain intact even when the need for circuit modification rises at a later stage.

[0158] **FIG. 25** shows a flow chart of one embodiment of the invention. The multiple-capture scan-test computer-aided design (CAD) system **2500** accepts the user-supplied HDL code or netlist **2502** together with the scan control files **2501** and the chosen foundry library **2503**. The scan control files **2501** contain all set-up information and scripts required for compilation **2504**, scan rule check **2506**, scan rule repair **2507**, and multiple-capture scan synthesis **2508**. As a result, an equivalent combinational circuit model **2509** is generated. Then, combinational ATPG **2510** can be performed. Finally, post-processing **2511** is used to produce the final scan HDL netlist **2512** as well as the HDL test benches and ATE test programs **2513**. All reports and errors are saved in the report files **2514**.

[0159] Having thus described presently preferred embodiments of the present invention, it can now be appreciated

that the objectives of the invention have been fully achieved. And it will be understood by those skilled in the art that many changes in construction & circuitry, and widely differing embodiments & applications of the invention will suggest themselves without departing from the spirit and scope of the present invention. The disclosures and the description herein are intended to be illustrative and are not in any sense limitation of the invention, more preferably defined in the scope of the invention by the Claims appended hereto and their equivalents.

What is claimed is:

1. A method for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in self-test mode, where N>1 and each domain has a plurality of scan cells, said method comprising the steps of:

(a) generating and loading N pseudorandom stimuli to all said scan cells within said N clock domains in said integrated circuit or circuit assembly during the shift operation;

(b) applying an ordered sequence of capture clocks to all said scan cells within said N clock domains during the capture operation;

(c) compacting N output responses of all said scan cells to signatures during the compact operation; and

(d) repeating the steps of (a)-(c) until a predetermined limiting criteria is reached, wherein (a) and (c) occur substantially concurrently.

2. The method of claim 1, wherein each said capture clock is programmable to contain one or more clock pulses for performing said shift/compact and capture operations on all said scan cells within one said clock domain; wherein said clock domain is solely controlled by said capture clock; and said capture clock can be either generated internally or controlled externally, and can operate either at its rated clock speed (at-speed) or at a selected clock speed.

3. The method of claim 1, further comprising providing N scan enable (SE) signals each within one said clock domain; wherein said SE signals are used to switch operations from shift/compact to capture, and vice versa; and further said SE signals can be generated internally or controlled externally, and are operated either at the rated clock speeds (at-speed) or at selected clock speeds.

4. The method of claim 3, wherein said providing N scan enable (SE) signals further comprises using one global scan enable (GSE) signal to drive said N scan enable (SE) signals; wherein said GSE signal is operated at a selected reduced clock speed.

5. The method of claim 1, wherein said generating and loading N pseudorandom stimuli further comprises operating all capture clocks at selected clock speeds or at the same clock speed, and when operated at the same clock speed, all said capture clocks are skewed so that at any given time only scan cells within one said clock domain are changing states to reduce power consumption.

6. The method of claim 1, further comprising the step of comparing said signatures with their expected signatures for error indication, after said predetermined limiting criteria is reached; wherein said step of comparing said signatures with their expected signatures further comprises comparing said signatures inside said integrated circuit or circuit assembly or shifting out said signatures for comparison in an ATE.

7. The method of claim 1, wherein said generating and loading N pseudorandom stimuli further comprises using a plurality of pseudorandom pattern generators (PRPGs) and phase shifters.

8. The method of claim 7, wherein each said pseudorandom pattern generator (PRPG) further comprises using a finite-state machine to automatically generate a number of test patterns; wherein said test patterns are applied through a phase shifter to a plurality of clock domains.

9. The method of claim 7, wherein each said phase shifter further comprises using a combinational logic network to decompress said test patterns to said pseudorandom stimuli.

10. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises performing said capture operation concurrently on a plurality of clock domains which do not have any logic block crossing each other.

11. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises applying said capture clocks in a selected order for detecting or locating additional faults in said integrated circuit or circuit assembly.

12. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises applying another ordered sequence of capture clocks selectively longer or shorter than said ordered sequence of capture clocks for detecting or locating additional faults in said integrated circuit or circuit assembly.

13. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises disabling one or more capture clocks to facilitate fault diagnosis.

14. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises selectively operating said capture clock at a selected clock speed for detecting or locating stuck-at faults within the clock domain controlled by said capture clock.

15. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises selectively operating said capture clock at its rated clock speed for detecting or locating delay faults within the clock domain controlled by said capture clock.

16. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises selectively reducing said capture clock speed to the level where delay faults associated with all multiple-cycle paths of equal cycle latency within the clock domain are tested at a predetermined rated clock speed.

17. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises selectively operating two said capture clocks at selected clock speeds for detecting or locating stuck-at faults crossing two said clock domains.

18. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises selectively adjusting the relative clock delay of two said capture clocks operating at selected clock speeds for detecting or locating delay faults crossing two said clock domains.

19. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises selectively adjusting the relative clock delay of two said capture clocks to the level where delay faults associated with all

multiple-cycle paths of equal cycle latency crossing two said clock domains are tested at a predetermined rated clock speed.

20. The method of claim 1, wherein said applying an ordered sequence of capture clocks further comprises controlling the relative clock delay between any two adjacent capture clocks internally or external to said integrated circuit or circuit assembly.

21. The method of claim 1, wherein said compacting N output responses further comprises using a plurality of space compactors and multiple-input signature registers (MISRs).

22. The method of claim 21, wherein each said space compactor further comprises using a combinational logic network to compress said output responses to compressed output responses.

23. The method of claim 21, wherein each said multiple-input signature register (MISR) further comprises using a finite-state machine to compact said compressed output responses to a signature; said MISR compacts said output responses through a space compactor to said signature.

24. The method of claim 1, further comprising using a PRPG-MISR pair to test faults within a plurality of clock domains when all capture clocks of said a plurality of clock domains operate at the same clock speed; all said capture clocks are skewed so as to eliminate races and timing violation during each shift, capture, or compact operation.

25. The method of claim 24, wherein said PRPG-MISR pair further comprises a PRPG, an optional phase shifter, an optional space compactor, a MISR, and a comparator.

26. The method of claim 25, wherein said PRPG-MISR pair further comprises connecting said PRPG to the first-arrived capture clock and connecting said MISR to the last-arrived capture clock within said a plurality of clock domains.

27. The method of claim 1, wherein said compacting N output responses further comprises selectively comparing said N output responses directly with their expected output responses and indicating errors immediately using a compare operation.

28. The method of claim 1, wherein said scan cells are multiplexed D flip-flops or level sensitive latches, and further wherein said integrated circuit or circuit assembly under test is a full-scan or partial-scan design.

29. The method of claim 1, wherein said faults further comprise stuck-at faults and delay faults; wherein said stuck-at faults further comprises other stuck-type faults, such as open and bridging faults, and wherein said delay faults further comprise other non-stuck-type delay faults, such as transition (gate-delay), multiple-cycle delay, and path-delay faults.

30. An apparatus for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in self-test mode, where N>1 and each domain has a plurality of scan cells, said apparatus comprising:

(a) means for generating and loading N pseudorandom stimuli to all said scan cells within said N clock domains in said integrated circuit or circuit assembly during the shift operation;

(b) means for applying an ordered sequence of capture clocks to all said scan cells within said N clock domains during the capture operation;

(c) means for compacting N output responses of all said scan cells to signatures during the compact operation; and

(d) means for repeating the steps of (a)-(c) until a predetermined limiting criteria is reached, wherein (a) and (c) occur substantially concurrently.

31. The apparatus of claim 30, wherein said means of (a)-(d) are placed inside or external to said integrated circuit or circuit assembly.

32. A method for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in self-test mode, where N>1 and each domain has a plurality of scan cells, said method comprising the steps of:

(a) shifting in N pseudorandom stimuli to all said scan cells within said N clock domains in said integrated circuit or circuit assembly during the shift-in operation;

(b) applying an ordered sequence of capture clocks to all said scan cells within said N clock domains during the capture operation; and

(c) shifting out N output responses of all said scan cells for analysis during the shift-out operation.

33. The method of claim 32, further comprising providing N scan enable (SE) signals each within one said clock domain; wherein said SE signals are used to switch operations from shift/compact to capture, and vice versa; and further said SE signals can be generated internally or controlled externally, and are operated either at the rated clock speeds (at-speed) or at selected clock speeds.

34. The method of claim 33, wherein said providing N scan enable (SE) signals further comprises using one global scan enable (GSE) signal to drive said N scan enable (SE) signals; wherein said GSE signal is operated at a selected reduced clock speed.

35. The method of claim 32, wherein said applying an ordered sequence of capture clocks further comprises any means for generating the ordered capture sequence; wherein said ordered capture sequence does not include any shift clock pulses during said capture operation.

36. A computer-aided design (CAD) system for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in self-test mode, where N>1 and each domain has a plurality of scan cells, said CAD system comprising the computer-implemented steps of:

(a) compiling the HDL code or netlist that represents said integrated circuit or circuit assembly in physical form into a design database;

(b) performing self-test rule check for checking whether said design database contains any multiple-capture self-test rule violations;

(c) performing self-test rule repair until all said multiple-capture self-test rule violations have been fixed;

(d) performing multiple-capture self-test synthesis for generating a self-test HDL code or netlist; and

(e) generating HDL test benches and ATE test programs for verifying the correctness of said self-test HDL code or netlist.

37. The CAD system of claim 36, wherein said steps of (a)-(e) accept user-supplied self-test control information and report the results and errors, if any.

38. The CAD system of claim 36, wherein said performing self-test rule check further comprises determining the number of clock domains and capture clocks required for self-test, the clock domains to be tested concurrently, the ordered sequence of capture clocks to be applied for self-test, and the capture clocks to be operated at the rated clock speeds or at selected clock speeds.

39. The CAD system of claim 36, wherein said performing multiple-capture self-test synthesis further comprises the hierarchical computer-implemented steps of:

(a) synthesizing a plurality of PRPG-MISR pairs one at a time for each individual clock domain or combined clock domains, wherein each said PRPG-MISR pair further comprises a PRPG, an optional phase shifter, an optional space compactor, a MISR, and a comparator;

(b) synthesizing a central self-test controller which includes an error indicator; and

(c) stitching said central self-test controller together with said PRPG-MISR pairs.

40. The CAD system of claim 39, wherein said synthesizing a plurality of PRPG-MISR pairs domain-by-domain further comprises inserting spare scan cells into selected clock domains.

41. The CAD system of claim 36, wherein said performing multiple-capture self-test synthesis realizes said apparatus of claim 30 using said method of claim 1.

42. The CAD system of claim 36, wherein said generating HDL test benches and ATE test programs further comprises the steps of transforming said design database into an equivalent combinational circuit model based on said ordered sequence of capture clocks, and performing combinational fault simulation to compute the circuit's output responses, signatures, and fault coverage.

43. A method for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in scan-test mode, where N>1 and each domain has a plurality of scan cells, said method comprising the steps of:

(a) generating and loading N predetermined stimuli to all said scan cells within said N clock domains in said integrated circuit or circuit assembly during the shift operation;

(b) applying an ordered sequence of capture clocks to all said scan cells within said N clock domains where one or more capture clocks must contain two or more clock pulses during the capture operation;

(c) comparing N output responses directly with their expected output responses for all said scan cells within said N clock domains and indicating errors immediately during the compare operation; and

(d) repeating the steps of (a)-(c) until a predetermined limiting criteria is reached, wherein (a) and (c) occur substantially concurrently.

44. The method of claim 43, wherein each said capture clock is programmable to contain one or more clock pulses for performing said shift/compare and capture operations on all said scan cells within one said clock domain; wherein said clock domain is solely controlled by said capture clock; and said capture clock can be either generated internally or controlled externally, and can operate either at its rated clock speed (at-speed) or at a selected clock speed.

45. The method of claim 43, further comprising providing N scan enable (SE) signals each within one said clock domain; wherein said SE signals are used to switch operations from shift/compact to capture, and vice versa; and further said SE signals can be generated internally or controlled externally, and are operated either at the rated clock speeds (at-speed) or at selected clock speeds.

46. The method of claim 45, wherein said providing N scan enable (SE) signals further comprises using one global scan enable (GSE) signal to drive said N scan enable (SE) signals; wherein said GSE signal is operated at a selected reduced clock speed.

47. The method of claim 43, wherein said generating and loading N predetermined stimuli further comprises operating all capture clocks at selected clock speeds or at the same clock speed, and when operated at the same clock speed, all said capture clocks are skewed so that at any given time only scan cells within one said clock domain are changing states to reduce power consumption.

48. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises performing said capture operation concurrently on a plurality of clock domains which do not have any logic block crossing each other.

49. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises applying said capture clocks in a selected order for detecting or locating additional faults in said integrated circuit or circuit assembly.

50. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises applying another ordered sequence of capture clocks selectively longer or shorter than said ordered sequence of capture clocks for detecting or locating additional faults in said integrated circuit or circuit assembly.

51. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises disabling one or more capture clocks to facilitate fault diagnosis.

52. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises selectively operating said capture clock at a selected clock speed for detecting or locating stuck-at faults within the clock domain controlled by said capture clock.

53. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises selectively operating said capture clock at its rated clock speed for detecting or locating delay faults within the clock domain controlled by said capture clock.

54. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises selectively reducing said capture clock speed to the level where delay faults associated with all multiple-cycle paths of equal cycle latency within the clock domain are tested at a predetermined rated clock speed.

55. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises selectively operating two said capture clocks at selected clock speeds for detecting or locating stuck-at faults crossing two said clock domains.

56. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises selectively adjusting the relative clock delay of two said capture clocks operating at selected clock speeds for detecting or locating delay faults crossing two said clock domains.

57. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises selectively adjusting the relative clock delay of two said capture clocks to the level where delay faults associated with all multiple-cycle paths of equal cycle latency crossing two said clock domains are tested at a predetermined rated clock speed.

58. The method of claim 43, wherein said applying an ordered sequence of capture clocks further comprises controlling the relative clock delay between any two adjacent capture clocks internally or external to said integrated circuit or circuit assembly.

59. The method of claim 43, wherein said comparing N output responses directly with their expected output responses further comprises selectively compacting said N output responses to signatures using a compact operation.

60. The method of claim 59, wherein said compacting N output responses to signatures further comprises comparing said signatures with their expected signatures after said predetermined limiting criteria is reached; wherein said comparing said signatures with their expected signatures further comprises comparing said signatures inside said integrated circuit or shifting out said signatures for comparison in an ATE.

61. The method of claim 43, wherein said scan cells are multiplexed D flip-flops or level sensitive latches, and further wherein said integrated circuit or circuit assembly under test is a full-scan or partial-scan design.

62. The method of claim 43, wherein said faults further comprise stuck-at faults and delay faults; wherein said stuck-at faults further comprises other stuck-type faults, such as open and bridging faults, and wherein said delay faults further comprises other non-stuck-type delay faults, such as transition (gate-delay), multiple-cycle delay, and path-delay faults.

63. An apparatus for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in scan-test mode, where N>1 and each domain has a plurality of scan cells, said apparatus comprising:

(a) means for generating and loading N predetermined stimuli to all said scan cells within said N clock domains in said integrated circuit or circuit assembly during the shift operation;

(b) means for applying an ordered sequence of capture clocks to all said scan cells within said N clock domains where one or more capture clocks must contain two or more clock pulses during the capture operation;

(c) means for comparing N output responses directly with their expected output responses for all said scan cells within said N clock domains and indicating errors immediately during the compare operation; and

(d) means for repeating the steps of (a)-(c) until a predetermined limiting criteria is reached, wherein (a) and (c) occur substantially concurrently.

64. The apparatus of claim 63, wherein said means of (a)-(d) are placed inside or external to said integrated circuit or circuit assembly.

65. A method for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in scan-test mode, where N>1 and each domain has a plurality of scan cells, said method comprising the steps of:

(a) shifting in N predetermined stimuli to all said scan cells within said N clock domains in said integrated circuit or circuit assembly during the shift-in operation;

(b) applying an ordered sequence of capture clocks to all said scan cells within said N clock domains where one or more capture clocks must contain two or more clock pulses during the capture operation; and

(c) shifting out N output responses of all said scan cells for analysis during the shift-out operation.

66. The method of claim 65, further comprising providing N scan enable (SE) signals each within one said clock domain; wherein said SE signals are used to switch operations from shift/compact to capture, and vice versa; and further said SE signals can be generated internally or controlled externally, and are operated either at the rated clock speeds (at-speed) or at selected clock speeds.

67. The method of claim 66, wherein said providing N scan enable (SE) signals further comprises using one global scan enable (GSE) signal to drive said N scan enable (SE) signals; wherein said GSE signal is operated at a selected reduced clock speed.

68. The method of claim 65, wherein said applying an ordered sequence of capture clocks further comprises any means for generating the ordered capture sequence; wherein said ordered capture sequence does not include any shift clock pulses during said capture operation.

69. A computer-aided design (CAD) system for providing ordered capture clocks to detect or locate faults within N clock domains and faults crossing any two clock domains in an integrated circuit or circuit assembly in scan-test mode, where N>1 and each domain has a plurality of scan cells, said CAD system comprising the computer-implemented steps of:

(a) compiling the HDL code or netlist that represents said integrated circuit or circuit assembly in physical form into a design database;

(b) performing scan rule check for checking whether said design database contains any multiple-capture scan rule violations;

(c) performing scan rule repair until all said multiple-capture scan rule violations have been fixed;

(d) performing multiple-capture scan synthesis for generating a scan HDL netlist; and

(e) generating HDL test benches and ATE test programs, where one or more capture clocks must contain two or more clock pulses, for verifying the correctness of said scan HDL netlist.

70. The CAD system of claim 69, wherein said steps of (a)-(e) accept user-supplied scan control information and report the results and errors, if any.

71. The CAD system of claim 69, wherein said performing scan rule check further comprises determining the number of clock domains and capture clocks required for scan-test, the clock domains to be tested concurrently, the ordered sequence of capture clocks to be applied for scan-test, and the capture clocks to be operated at the rated clock speeds or at selected clock speeds.

72. The CAD system of claim 69, wherein said performing multiple-capture scan synthesis further comprises inserting spare scan cells into selected clock domains.

73. The CAD system of claim 69, wherein said performing multiple-capture scan synthesis realizes said apparatus of claim 63 using said method of claim 43.

74. The CAD system of claim 69, wherein said generating HDL test benches and ATE test programs further comprises the steps of transforming said design database into an equivalent combinational circuit model based on said ordered sequence of capture clocks, and performing combinational ATPG to generate the circuit's test patterns and report its fault coverage.

75. The CAD system of claim 69, wherein said generating HDL test benches and ATE test programs further comprises performing combinational logic simulation on said combinational circuit model to compute said circuit's signatures when a compact operation is employed to compact said circuit's output responses.

* * * * *

Lattice Semiconductor Corporation

Data Sheet
January, 2003

ORCA® Series 4 FPGAs

Introduction

Built on the Series 4 reconfigurable embedded system-on-a-chip (SoC) architecture, Lattice introduces its new family of generic Field-Programmable Gate Arrays (FPGAs). The high-performance and highly versatile architecture brings a new dimension to bringing network system designs to market in less time than ever before. This new device family offers many new features and architectural enhancements not available in any earlier FPGA generations. Bringing together highly flexible SRAM-based programmable logic, powerful system features, a rich hierarchy of routing and interconnect resources, and meeting multiple interface standards, the Series 4 FPGA accommodates the most complex and high-performance intellectual property (IP) network designs.

Programmable Features

- High-performance platform design:
  - 0.16 µm 7-level metal technology.
  - Internal performance of >250 MHz.
  - I/O performance of >420 MHz.
  - Meets multiple I/O interface standards.
  - 1.5 V operation (30% less power than 1.8 V operation) translates to greater performance.
- Traditional I/O selections:
  - LVTTL (3.3V) and LVCMOS (2.5 V and 1.8 V) I/Os.
  - Per pin-selectable I/O clamping diodes provide 3.3 V PCI compliance.
  - Individually programmable drive capability: 24 mA sink/12 mA source, 12 mA sink/6 mA source, or 6 mA sink/3 mA source.
  - Two slew rates supported (fast and slew-limited).
  - Fast-capture input latch and input flip-flop (FF)/latch for reduced input setup time and zero hold time.
  - Fast open-drain drive capability.
  - Capability to register 3-state enable signal.
  - Off-chip clock drive capability.
  - Two-input function generator in output path.
- New programmable high-speed I/O:
  - Single-ended: GTL, GTL+, PECL, SSTL3/2 (class I and II), HSTL (Class I, III, and IV), ZBT, and DDR.
  - Double-ended: LDVS, bused-LVDS, and LVPECL. Programmable (on/off) internal parallel termination (100 Ω) also supported for these I/Os.

Table 1. *ORCA* Series 4—Available FPGA Logic

Device	Rows	Columns	PFUs	User I/O	LUTs	EBR Blocks	EBR Bits (K)	Usable* Gates (K)
OR4E02	26	24	624	405	4,992	8	74	201—397
OR4E04	36	36	1,296	466	10,368	12	111	333—643
OR4E06	46	44	2,024	466	16,192	16	148	471—899

* The embedded system bus and MPI are not included in the above gate counts. The System Gate ranges are derived from the following: minimum system gates assumes 100% of the PFUs are used for logic only (no PFU RAM) with 40% EBR usage and 2 PLLs. Maximum system gates assumes 80% of the PFUs are for logic, 20% are used for PFU RAM, with 80% EBR usage and 6 PLLs.

Note: Devices are not pinout compatible with *ORCA* Series 2/3.

Programmable Logic Cells (continued)

Look-Up Table Operating Modes

The operating mode affects the functionality of the PFU input and output ports and internal PFU routing. For example, in some operating modes, the DIN[7:0] inputs are direct data inputs to the PFU latches/FFs. In memory mode, the same DIN[7:0] inputs are used as a 4-bit write data input bus and a 4-bit write address input bus into LUT memory.

Table 2 lists the basic operating modes of the LUT. Figure 4—Figure 7 show block diagrams of the LUT operating modes. The accompanying descriptions demonstrate each mode's use for generating logic.

Table 2. Look-Up Table Operating Modes

Mode	Function
Logic	4-, 5-, and 6-input LUTs; softwired LUTs; latches/FFs with direct input or LUT input; CIN as direct input to ninth FF or as pass through to COUT.
Half Logic/ Half Ripple	Upper four LUTs and latches/FFs in logic mode; lower four LUTs and latches/FFs in ripple mode; CIN and ninth FF for logic or ripple functions.
Ripple	All LUTs combined to perform ripple-through data functions. Eight LUT registers available for direct-in use or to register ripple output. Ninth FF dedicated to ripple out, if used. The submodes of ripple mode are adder/subtractor, counter, multiplier, and comparator.
Memory	All LUTs and latches/FFs used to create a 32x4 synchronous dual-port RAM. Can be used as single-port or as ROM.

PFU Control Inputs

Each PFU has eight routable control inputs and an active-low, asynchronous global set/reset (GSRN) signal that affects all latches and FFs in the device. The eight control inputs are CLK[1:0], LSR[1:0], CE[1:0], and SEL[1:0], and their functionality for each logic mode of the PFU is shown in Table 3. The clock signal to the PFU is CLK, CE stands for clock enable, which is its primary function. LSR is the local set/reset signal that can be configured as synchronous or asynchronous. The selection of set or reset is made for each latch/FF and is not a function of the signal itself. SEL is used to dynamically select between direct PFU input and LUT output data as the input to the latches/FFs.

All of the control signals can be disabled and/or inverted via the configuration logic. A disabled clock enable indicates that the clock is always enabled. A disabled LSR indicates that the latch/FF never sets/resets (except from GSRN). A disabled SEL input indicates that DIN[7:0] PFU inputs are routed to the latches/FFs.

Table 3. Control Input Functionality

Mode	CLK[1:0]	LSR[1:0]	CE[1:0]	SEL[1:0]
Logic	CLK to all latches/ FFs	LSR to all latches/FFs, enabled per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	Select between LUT input and direct input for eight latches/FFs
Half Logic/ Half Ripple	CLK to all latches/ FFs	LSR to all latches/FF, enabled per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	Select between LUT input and direct input for eight latches/FFs
Ripple	CLK to all latches/ FFs	LSR to all latches/FFs, enabled per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	Select between LUT input and direct input for eight latches/FFs
Memory (RAM)	CLK to RAM	LSR0 Port enable 2	CE1 RAM write enable CE0 Port enable 1	Not used
Memory (ROM)	Optional for synchronous outputs	Not used	Not used	Not used

Programmable Logic Cells (continued)

The set/reset operation of the latch/FF is controlled by two parameters: reset mode and set/reset value. When the GSRN and local set/reset (LSR) signals are not asserted, the latch/FF operates normally. The reset mode is used to select a synchronous or asynchronous LSR operation. If synchronous, LSR has the option to be enabled only if clock enable (CE) is active or for LSR to have priority over the clock enable input, thereby setting/resetting the FF independent of the state of the clock enable. The clock enable is supported on FFs, not latches. It is implemented by using a 2-input multiplexer on the FF input, with one input being the previous state of the FF and the other input being the new data applied to the FF. The select of this 2-input multiplexer is clock enable (CE), which selects either the new data or the previous state. When the clock enable is inactive, the FF output does not change when the clock edge arrives.

The GSRN signal is only asynchronous, and it sets/resets all latches/FFs in the FPGA based upon the set/reset configuration bit for each latch/FF. The set/reset value determines whether GSRN and LSR are set or reset inputs. The set/reset value is independent for each latch/FF. An option is available to disable the GSRN function per PFU after initial device configuration.

The latch/FF can be configured to have a data front-end select. Two data inputs are possible in the front-end select mode, with the SEL signal used to select which data input is used. The data input into each latch/FF is from the output of its associated LUT, F[7:0], or direct from DIN[7:0], bypassing the LUT. In the front-end data select mode, both signals are available to the latches/FFs.

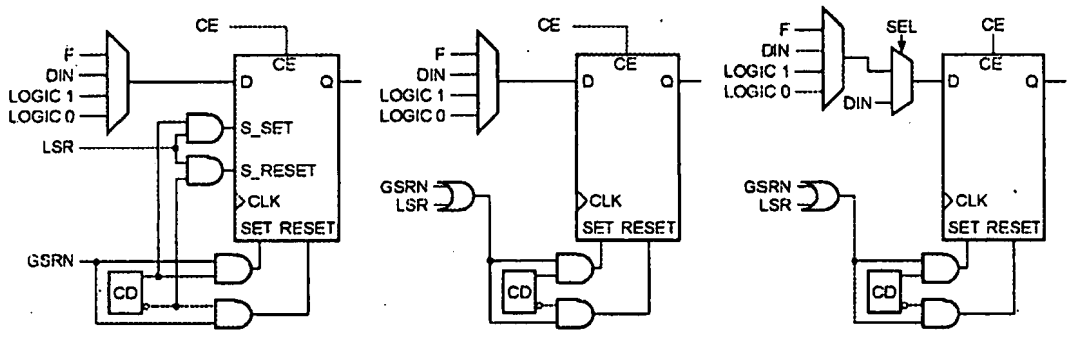
If either or both of these inputs is unused or is unavailable, the latch/FF data input can be tied to a logic 0 or logic 1 instead (the default is logic 0).

The latches/FFs can be configured in three basic modes:

- Local synchronous set/reset: the input into the PFU's LSR port is used to synchronously set or reset each latch/FF.
- Local asynchronous set/reset: the input into LSR asynchronously sets or resets each latch/FF.
- Latch/FF with front-end select, LSR either synchronous or asynchronous: the data select signal selects the input into the latches/FFs between the LUT output and direct data in.

For all three modes, each latch/FF can be independently programmed as either set or reset. Figure 20 provides the logic functionality of the front-end select, global set/reset, and local set/reset operations.

The ninth PFU FF, which is generally associated with registering the carry-out signal in ripple mode functions, can be used as a general-purpose FF. It is only an FF and is not capable of being configured as a latch. Because the ninth FF is not associated with an LUT, there is no front-end data select. The data input to the ninth FF is limited to the CIN input, logic 1, logic 0, or the carry-out in ripple and half-logic modes.



Key: C = configuration data.

Figure 20. Latch/FF Set/Reset Configurations

Routing Resources (continued)

Cycle Stealing

A new feature in Series 4 FPGAs is the ability to steal time from one register-to-register path and use that time in either the previous path before the first register or in a later path after the last register. This is done through selectable clock delays for every PLC register, EBR register, and PIO register. There are four programmable delay settings, including the default zero added delay value. This allows performance increases on typical critical paths from 15% to 40%. ispLEVER includes software to automatically take advantage of this capability to increase overall system speed. This is done after place and route is completed and uses timing driven algorithms based on the customer's preference file. A hold time check is also performed to verify no minimum hold time issues are introduced. More information on this clocking feature, including how it can be used to improve device setup times, hold times, clock-to-out delays and can reduce ground bounce caused by switching outputs can be found in the Cycle Stealing application note.

Programmable Input/Output Cells (PIC)

Programmable I/O

The Series 4 programmable I/O addresses the demand for the flexibility to select I/O that meets system interface requirements. I/Os can be programmed in the same manner as in previous *ORCA* devices with the addition of new features which allow the user the flexibility to select new I/O types that support high-speed interfaces.

Each PIC contains up to four programmable I/O (PIO) pads and are interfaced through a common interface block (CIB) to the FPGA array. The PIC is split into two pairs of I/O pads with each pair having independent clocks, clock enables, local set/reset, and global set/reset enable/disable.

On the input side, each PIO contains a programmable latch/FF which enables very fast latching of data from any pad. The combination provides for very low setup requirements and zero hold times for signals coming on-chip. It may also be used to demultiplex an input signal, such as a multiplexed address/data signal, and register the signals without explicitly building a demultiplexer with a PFU.

On the output side of each PIO, an output from the PLC array can be routed to each output FF, and logic can be associated with each I/O pad. The output logic associated with each pad allows for multiplexing of output signals and other functions of two output signals.

The output FF, in combination with output signal multiplexing, is particularly useful for registering address signals to be multiplexed with data, allowing a full clock cycle for the data to propagate to the output. The output buffer signal can be inverted, and the 3-state control can be made active-high, active-low, or always enabled. In addition, this 3-state signal can be registered or nonregistered.

The Series 4 I/O logic has been enhanced to include modes for high-speed uplink and downlink capabilities. These modes are supported through shift register logic which divides down incoming data or multiplies up outgoing data. This new logic block also supports high-speed DDR mode requirements where data is clocked into and out of the I/O buffers on both edges of the clock.

The new programmable I/O cell allows designers to select I/Os which meet many new communication standards permitting the device to hook up directly without any external interface translation. They support traditional FPGA standards as well as high-speed single-ended and differential pair signaling (as shown in Table 13). Based on a programmable, bank-oriented I/O ring architecture, designs can be implemented using 3.3 V, 2.5 V, 1.8 V, and 1.5 V I/O levels.

The I/O on the OR4Exx Series devices allows compliance with PCI local bus (Rev. 2.2) 3.3 V signaling environments. The signaling environment used for each input buffer can be selected on a per-pin basis. The selection provides the appropriate I/O clamping diodes for PCI compliance.

More information on the Series 4 programmable I/O structure is available in the various application notes.

Programmable Input/Output Cells

(continued)

Inputs

There are many major options on the PIO inputs that can be selected in the ispLEVER tools listed in Table 14. Inputs may have a pull-up or pull-down resistor selected on an input for signal stabilization and power management. Input signals in a PIO are passed to CIB routing and/or a fast route into the clock routing system. A fast input from one PIO per PIC is also available to drive the edge clock network for fast I/O timing to other nearby PIOs.

There is also a programmable delay available on the input. When enabled, this delay affects the INFF and INDD signals of each PIO, but not the clock input. The delay allows any signal to have a guaranteed zero hold time when input.

Inputs should have transition times of less than 100 ns and should not be left floating. For full swing inputs, the timing characterization is done for rise/fall times of ≥ 1 V/ns. If any pin is not used, it is 3-stated with an internal pull-up resistor enabled automatically after configuration. Floating inputs increase power consumption, produce oscillations, and increase system noise. The inputs in LVTTL, LVCMOS2, and LVCMOS18 modes have a typical hysteresis of approximately 250 mV to reduce sensitivity to input noise. The PIC contains input circuitry which provides protection against latch-up and electrostatic discharge.

The other features of the PIO inputs relate to the latch/FF structure in the input path. In latch mode, the input signal is fed to a latch that is clocked by either the primary, secondary, or edge clock signal. The clock may be inverted or noninverted. There is also a local set/reset signal to the latch. The senses of these signals are also programmable as well as the capability to enable or disable the global set/reset signal and select the set/reset priority. The same control signals may also be used to control the input latch/FF when it is configured as a FF instead of a latch, with the addition of another control signal used as a clock enable. The PIOs are paired together and have independent CE, Set/reset, and GSRN control signals per PIO pair.

There are two options for zero-hold input capture in the PIO. If input delay mode is selected to delay the signal from the input pin, data can be either registered or latched with guaranteed zero-hold time in the PIO using a global primary system clock. The fast zero-hold mode of the PIO input takes advantage of a latch/FF combination to latch the data quickly for zero-hold using a fast edge clock before passing the data to the FF which is clocked by a global primary system clock.

The combination of input register capability with non-registered inputs provides for input signal demultiplexing without any additional resources. The PIO input signal is sent to both the input register and directly to the unregistered input (INDD). The signal is latched and output to routing at INFF. These signals may then be registered or otherwise processed in the PLCs.

Every PIO input can also perform input double data rate (DDR) functions with no PLC resources required. This type of scheme is necessary for DDR applications which require data to be clocked in from the I/O on both edges of the clock. In this scheme the input of INFF and INSH are captured on the positive and negative edges of the clock.

Table 14. PIO Options

Input	Option
Input Speed	Fast, Delayed, Normal
Float Value	Pull-up, Pull-down, None
Register Mode	Latch, FF, Fast Zero Hold FF, None (direct input)
Clock Sense	Inverted, Noninverted
Keeper Mode	on, off
LVDS Resistor	on, off
Output	**Option**
Output Speed	Fast, Slew
Output Drive Current	12 mA/6 mA, 6 mA/3 mA, or 24 mA/12 mA
Output Function	Normal, Fast Open Drain
Output Sense	Active-high, Active-low
3-State Sense	Active-high, Active-low
Clock Sense	Inverted, Noninverted
Logic Options	See Table 15
I/O Controls	**Option**
Clock Enable	Active-high, Active-low, Always Enabled
Set/Reset Level	Active-high, Active-low, No Local Reset
Set/Reset Type	Synchronous, Asynchronous
Set/Reset Priority	CE over LSR, LSR over CE
GSR Control	Enable GSR, Disable GSR

Programmable Input/Output Cells
(continued)

Outputs

The PIO's output drivers have programmable drive capability and slew rates. Two propagation delays (fast, slewlim) are available on output drivers. There are three combinations of programmable drive currents (24 mA sink/12 mA source, 12 mA sink/6 mA source, and 6 mA sink/3 mA source). At powerup, the output drivers are in slewlim mode with 12mA sink/6 mA source. If an output is not to be driven in the selected configuration mode, it is 3-stated with a pullup resistor.

The output buffer signal can be inverted, and the 3-state control signal can be made active-high, active-low, or always enabled. In addition, this 3-state signal can be registered or nonregistered. Additionally, there is a fast, open-drain output option that directly connects the output signal to the 3-state control, allowing the output buffer to either drive to a logic 0 or 3-state, but never to drive to a logic 1.

Every PIO output can perform output data multiplexing with no PLC resources required. This type of scheme is necessary for DDR applications which require data clocking out of the I/O on both edges of the clock. In this scheme the OUTFF and OUTSH are registered and sent out on both the positive and negative edges of the clock using an output multiplexor. This multiplexor is controlled by either the edge clock or system clock. This multiplexor can also be configured to select between one registered output from OUTFF and one nonregistered output from OUTDD.

The PIC logic block can also generate logic functions based on the signals on the OUTDD and CLK ports of the PIO. The functions are AND, NAND, OR, NOR, XOR, and XNOR. Table 15 is provided as a summary of the PIO logic options.

Table 15. PIO Logic Options

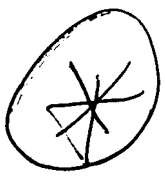
Option	Description
AND	Output logical AND of signals on OUTDD and clock.
NAND	Output logical NAND of signals on OUTDD and clock.
OR	Output logical OR of signals on OUTDD and clock.
NOR	Output logical NOR of signals on OUTDD and clock.
XOR	Output logical XOR of signals on OUTDD and clock.
XNOR	Output logical XNOR of signals on OUTDD and clock.

PIO Register Control Signals

The PIO latches/FFs have various clock, clock enable (CE), local set/reset (LSR), and GSRN controls. Table 16 provides a summary of these control signals and their effect on the PIO latches/FFs. Note that all control signals are optionally invertible.

Table 16. PIO Register Control Signals

Control Signal	Effect/Functionality
Edge Clock (ECLK)	Clocks input fast-capture latch; optionally clocks output FF, or 3-state FF, or PIO shift registers.
System Clock (SCLK)	Clocks input latch/FF; optionally clocks output FF, or 3-state FF, or PIO shift registers.
Clock Enable (CE)	Optionally enables/disables input FF (not available for input latch mode); optionally enables/disables output FF; separate CE inversion capability for input and output.
Local Set/ Reset (LSR)	Option to disable; affects input latch/FF, output FF, and 3-state FF if enabled.
Global Set/ Reset (GSRN)	Option to enable or disable per PIO after initial configuration.
Set/Reset Mode	The input latch/FF, output FF, and 3-state FF are individually set or reset by both the LSR and GSRN inputs.



Programmable Input/Output Cells

(continued)

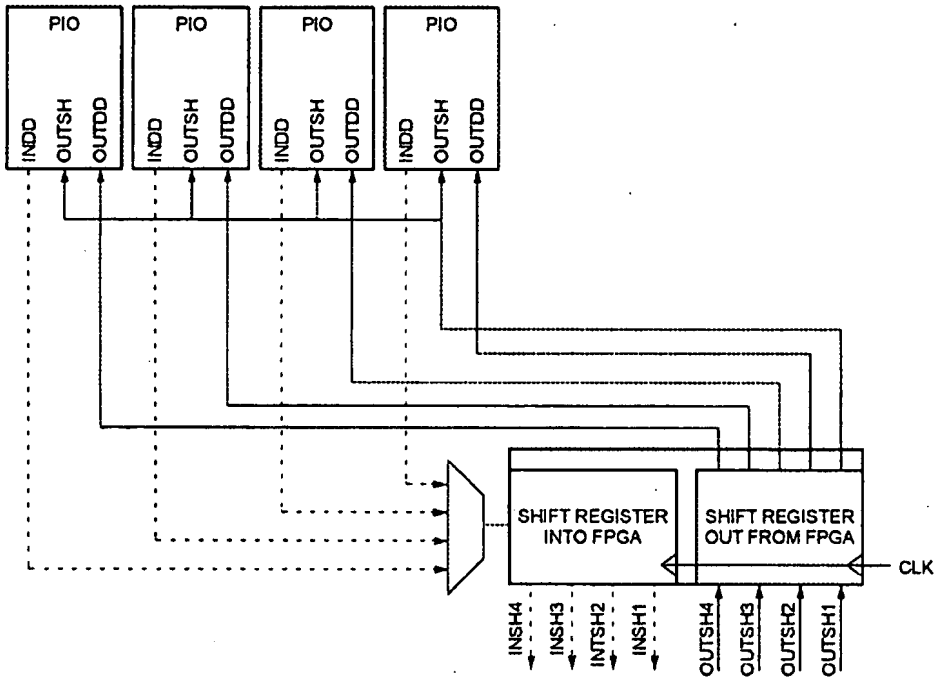


Figure 24. PIO Shift Register

Special Function Blocks

Special function blocks in the Series 4 provide extra capabilities beyond general FPGA operation. These blocks reside in the corners and MIDs (middle inter-quad areas) of the FPGA array.

Internal Oscillator

The internal oscillator resides in the upper left corner of the FPGA array. It has output clock frequencies of 1.25 MHz and 10 MHz. The internal oscillator is the source of the internal CCLK used for configuration. It may also be used after configuration as a general-purpose clock signal.

Global Set/Reset (GSRN)

The GSRN logic resides in the upper-left corner of the FPGA. GSRN is an invertible, default, active-low signal that is used to reset all of the user-accessible latches/FFs on the device. GSRN is automatically asserted at powerup and during configuration of the device.

The timing of the release of GSRN at the end of configuration can be programmed in the start-up logic described below. Following configuration, GSRN may be connected to the $\overline{\text{RESET}}$ pin via dedicated routing, or it may be connected to any signal via normal routing. GSRN can also be controlled via a system bus register command. Within each PFU and PIO, individual FFs and latches can be programmed to either be set or reset when GSRN is asserted. Series 4 allows individual PFUs and PIOs to turn off the GSRN signal to its latches/FFs after configuration.

The $\overline{\text{RESET}}$ input pad has a special relationship to GSRN. During configuration, the $\overline{\text{RESET}}$ input pad always initiates a configuration abort, as described in the FPGA States of Operation section. After configuration, the GSRN can either be disabled (the default), directly connected to the $\overline{\text{RESET}}$ input pad, or sourced by a lower-right corner signal. If the $\overline{\text{RESET}}$ input pad is not used as a global reset after configuration, this pad can be used as a normal input pad.

FPGA States of Operation (continued)

Power Supply Sequencing

FPGAs are CMOS static RAM (SRAM) based programmable logic devices. The circuitry that the user designs for the FPGA is implemented within the FPGA by setting multiple SRAM configuration memory cells. This unique structure as compared with typical CMOS circuits lends to having certain powerup voltage and current requirements. This section describes these related power issues for the *ORCA* Series 4 FPGAs and FPSCs.

The flexibility of Series 4 FPGAs lends itself to more power up considerations as it mixes many power supplies to meet today's versatile system standards. The board designer must account for the relationship of the supplies early in board development. The proper sequence of supplies insures that the board will not be troubled with power up issues.

The Series 4 devices have many new design improvements to prevent short-circuit contention. This contention is typically caused by configuration RAM cells in the device not all powering up to a Q = 0 RAM state. In order for this to occur, a minimum current was needed to push the internal circuitry beyond the initial short-circuit-like condition to become a full CMOS circuit. Series 4 has overcome this requirement through many improvements which have dramatically decreased the adverse effects of internal power up memory contention.

At power up, the internal V_{DD} ramp and the duration of the ramp will depend on the amount of dynamic current available from the power supply. If a large amount of current is available, the voltage ramp seen by the device will be very fast. When final voltage has been reached, this high quiescent current is no longer required. If the available current is limited, the time for the device power to rise will be longer. The voltage ramp should be monotonic with very little or no flattening as the supply ramps up. It is also recommended that the supply should not rise and fall as it is powering up as this will cause improper power up behavior.

In Series 4 devices, it is recommended that the $V_{DD}15$ supply pass through its operational threshold voltage of approximately 1 V before the $V_{DD}33$ supply reaches its operational threshold of 2.3 V. The current required by both $V_{DD}15$ and $V_{DD}33$ supplies while it passes through their operational thresholds is approximately between 1 and 2 amperes each. The powering of the $V_{DD}IO$ supplies should be after the $V_{DD}15$ and $V_{DD}33$ supplies reach operational levels. This sequence and supply currents can guarantee that the device will properly power up without any adverse effects.

In cases where the power up ramps are greater than 50 mS, it is recommended that $\overline{PRGM}$ pin be held low during power up. However, this work around is only valid if the power supplies meet the above mentioned current and voltage requirements. The assertion of the $\overline{PRGM}$ will hold off the device from configuration while the device stabilizes and will not counter act any internal power up requirements.

Configuration

The *ORCA* Series FPGA functionality is determined by the state of internal configuration RAM. This configuration RAM can be loaded in a number of different modes. In these configuration modes, the FPGA can act as a master or a slave of other devices in the system. The decision as to which configuration mode to use is a system design issue. Configuration is discussed in detail, including the configuration data format and the configuration modes used to load the configuration data in the FPGA, following a description of the start-up state.

Start-Up

After configuration, the FPGA enters the start-up phase. This phase is the transition between the configuration and operational states and begins when the number of CCLKs received after $\overline{INIT}$ goes high is equal to the value of the length count field in the configuration frame and when the end of configuration frame has been written. The system design issue in the start-up phase is to ensure the user I/Os become active without inadvertently activating devices in the system or causing bus contention. A second system design concern is the timing of the release of global set/reset of the PLC latches/FFs.

FPGA States of Operation (continued)

Reconfiguration

To reconfigure the FPGA when the device is operating in the system, a low pulse is input into $\overline{\text{PRGM}}$ or one of the program bits in the embedded system bus control register must be set. The configuration data in the FPGA is cleared, and the I/Os not used for configuration are 3-stated with a pullup. The FPGA then samples the mode select inputs and begins reconfiguration. When reconfiguration is complete, DONE is released, allowing it to be pulled high.

Partial Reconfiguration

All *ORCA* device families have been designed to allow a partial reconfiguration of the FPGA at any time. This is done by setting a bit stream option in the previous configuration sequence that tells the FPGA to not reset all of the configuration RAM during a reconfiguration. Then only the configuration frames that are to be modified need to be rewritten, thereby reducing the configuration time.

Other bit stream options are also available that allow one portion of the FPGA to remain in operation while a partial reconfiguration is being done. If this is done, the user must be careful to not cause contention between the two configurations (the bit stream resident in the FPGA and the partial reconfiguration bit stream) as the second reconfiguration bit stream is being loaded.

During a partial re-configuration where the configuration option is set to have the internal logic remain active during configuration the internal SLJC BIDI signals will always be 3-stated. Previous families of ORCA FPGAs would allow the BIDIs to continue to be under user logic control during a partial re-configuration.

Other Configuration Options

There are many other configuration options available to the user that can be set during bit stream generation in ispLEVER. These include options to enable boundary-scan and/or the MPI and/or the programmable PLL blocks, readback options, and options to control and use the internal oscillator after configuration.

Other useful options that affect the next configuration (not the current configuration process) include options to disable the global set/reset during configuration, disable the 3-state of I/Os during configuration, and disable the reset of internal RAMs during configuration to allow for partial configurations (see above). For more information on how to set these and other configuration options, please see the ispLEVER documentation.

Configuration Data Format

The ispLEVER Development System interfaces with front-end design entry tools and provides tools to produce a fully configured FPGA. This section discusses using the ispLEVER Development System to generate configuration RAM data and then provides the details of the configuration frame format.

Using ispLEVER to Generate Configuration RAM Data

The configuration data bit stream defines the I/O functionality, logic, and interconnections within the FPGA. The bit stream is generated by the development system. The bit stream created by the bit stream generation tool is a series of 1s and 0s used to write the FPGA configuration RAM. It can be loaded into the FPGA using one of the configuration modes discussed later.

In bit stream generator, the designer selects options that affect the FPGA's functionality. Using the output of the bit stream generator, **circuit_name.bit**, the development system's download tool can load the configuration data into the *ORCA* series FPGA evaluation board from a PC or workstation.

A download cable that can be used to download from any PC or workstation supported by ispLEVER is available. This cable allows download to an FPGA that can be programmed via the serial configuration interface (requiring the mode pins to be set) or the JTAG boundary scan interface (not requiring the setting of mode pins). The lead device can then program other FPGAs or FPSCs on the board via daisy-chaining.

Alternatively, a user can program a PROM (such as a Serial ROM or a standard EPROM) and load the FPGA from the PROM. The development system's PROM programming tool produces a file in .mcs, .tek or .exo format.

[PDF] ORCA Series 4 FPGAs
File Format: PDF/Adobe Acrobat - View as HTML
Page 1. Data Sheet January, 2003 www.latticesemi.com ORCA ® Series 4 FPGAs
Introduction Built on the Series 4 reconfigurable embedded ...
www.pmaster.com.tw/lattice_pdf/Datasheet/FPGA/ORCA4.pdf - Supplemental Result - Similar pages